



Πανεπιστήμιο Πατρών
Πολυτεχνική Σχολή
Τμήμα Μηχανικών Η/Υ και
Πληροφορικής

UDP

Εργασία για το μάθημα

Δίκτυα Δημόσιας Χρήσης και Διασύνδεση Δικτύων

Διδάσκων:

Μπούρας Χρήστος

Ομάδα:

Αργυροπούλου Ευαγγελία, Α.Μ. 4152

Σαλτογιάννη Αθανασία, Α.Μ. 4255

Πάτρα, Δεκέμβριος 2011

Περιεχόμενα

1. Πρωτόκολλα επικοινωνίας

1.1. Ορισμός.....	4
1.2. Βασικές λειτουργίες.....	4
1.3. Ιεραρχία.....	5
1.4. Αρχή της διαστρωμάτωσης πρωτοκόλλων.....	6

2. Το μοντέλο αναφοράς OSI

2.1. Τι είναι το OSI.....	8
2.2. Επίπεδα μοντέλου OSI.....	9
2.3. Πρωτόκολλα μεταφοράς (Transport protocols).....	10
2.4. Μετάδοση βέλτιστης προσπάθειας.....	11
2.5. Θύρες (ports).....	12

3. Σουίτα TCP/IP

3.1. Τι είναι το TCP/IP.....	12
3.2. Internet Protocol – IP.....	14
3.3. Transmission Control Protocol – TCP.....	15
3.3.1.Εισαγωγή.....	15
3.3.2.Χαρακτηριστικά του TCP.....	15
3.3.3.Κεφαλίδα του TCP.....	16

4. Πρωτόκολλο UDP

4.1. Αναγνώριση του απόλυτου προορισμού.....	17
4.2. Τι είναι το UDP.....	19
4.3. Χαρακτηριστικά.....	20
4.4. Γιατί υπάρχει το UDP.....	21

5. Λειτουργικές προδιαγραφές του UDP

5.1. Μορφή μηνύματος του UDP.....	22
5.2. Ψευδοκεφαλίδα UDP.....	23

5.3. Ενθυλάκωση και διαστρωμάτωση του UDP.....	25
5.4. Πολύπλεξη και αποπολύπλεξη UDP.....	26
5.5. Υπολογισμός αθροίσματος ελέγχου UDP.....	27
5.5.1.IPv4.....	28
5.5.2.IPv6.....	29
5.6. Αριθμοί θυρών UDP.....	30

6. UDP server και client

6.1. Εισαγωγή στην έννοια client-server computing.....	32
6.2. Βασικό client-server μοντέλο.....	32
6.2.1.Client.....	33
6.2.2.Server.....	33
6.2.3.Δίκτυα.....	34
6.3. Προγραμματισμός με το UDP.....	34
6.4. Δείγματα απλού κώδικα.....	35
6.4.1.UDP server-client σε C.....	35
6.4.2.UDP server-client σε Java.....	38

7. Σύγκριση UDP με TCP.....39

8. Επίλογος-Περίληψη.....41

9. Γλωσσάρι όρων και συντομογραφιών.....43

10. Βιβλιογραφία46

1. Πρωτόκολλα επικοινωνίας

1.1. Ορισμός

Ως πρωτόκολλο επικοινωνίας ορίζεται ένα σύνολο κανόνων συμφωνημένων και από τα δύο επικοινωνούντα μέρη και που καθορίζουν την αποδοτική επικοινωνία μεταξύ δύο μερών. Δηλαδή, το πρωτόκολλο επικοινωνίας είναι μία δέσμη κανόνων στους οποίους στηρίζεται η επικοινωνία των υπολογιστών σε ένα δίκτυο. Οι κανόνες αυτοί καθορίζουν τη μορφή, το χρόνο και την σειρά μετάδοσης των πληροφοριών στο δίκτυο. Επίσης, εκτελούν έλεγχο και διόρθωση σφαλμάτων στη διάρκεια μετάδοσης των πληροφοριών.

Τα πρωτόκολλα δίνουν τη δυνατότητα σε κάποιον να ορίσει ή να κατανοήσει τη διαδικασία της επικοινωνίας χωρίς να γνωρίζει τις λεπτομέρειες του υλικού δικτύων ενός συγκεκριμένου κατασκευαστή. Θα μπορούσαμε να πούμε πως ό, τι είναι για την υπολογιστική επιστήμη οι γλώσσες προγραμματισμού είναι και τα πρωτόκολλα για την επικοινωνία των υπολογιστών.

Τα περίπλοκα συστήματα επικοινωνίας δεδομένων δεν χρησιμοποιούν ένα μόνο πρωτόκολλο για όλες τις εργασίες μετάδοσης. Αντιθέτως, χρειάζονται ένα σύνολο συνεργαζόμενων πρωτοκόλλων, το οποίο καλείται και οικογένεια πρωτοκόλλων ή συλλογή πρωτοκόλλων, ώστε να ξεπεραστούν προβλήματα που προκύπτουν όταν επικοινωνούν μηχανές μέσω ενός δικτύου δεδομένων όπως είναι η βλάβη υλικού, η καθυστέρηση ή απώλεια πακέτων και η αλλοίωση δεδομένων.

Υπάρχουν διάφορα πρωτόκολλα επικοινωνίας, τα οποία σήμερα προκαλούν πολλές φορές σύγχυση στους χρήστες. Σήμερα, παρόλο που δεν υπάρχει κάποιο που να είναι καθιερωμένο πρότυπο, τα πρωτόκολλα που είναι περισσότερο διαδεδομένα είναι το TCP/IP, UDP, NETBEUI και το IPX/SPX.

Σημείωση:

Οι σουίτες πρωτοκόλλων ονομάζονται στοίβες, επειδή περιλαμβάνουν μεμονωμένα πρωτόκολλα που δουλεύουν σε διαφορετικά επίπεδα του μοντέλου OSI. Για αυτό θα μιλήσουμε σε επόμενο κεφάλαιο.

1.2. Βασικές Λειτουργίες

Οι βασικές λειτουργίες ενός πρωτοκόλλου επικοινωνιών είναι:

- **Κατακερματισμός μηνυμάτων (segmentation)**
Ο χωρισμός του προς αποστολή μηνύματος σε μικρότερα τμήματα (blocks). Στόχος του κατακερματισμού είναι ο έλεγχος σφαλμάτων και να καλύπτει

καλύτερη ταχύτητα, γιατί σε περίπτωση σφάλματος στέλνεται ξανά μόνο ένα τμήμα και όχι ολόκληρο το μήνυμα.

- **Επανασύνδεση μηνυμάτων (reassembly)**
Επανασύνδεση του μηνύματος από τα επιμέρους τμήματα
- **Ενθυλάκωση μηνυμάτων (encapsulation)**
Τοποθέτηση των δεδομένων σε ένα πλαίσιο το οποίο περιέχει μεταξύ άλλων τις διευθύνσεις αποστολής και παραλήπτη. Χαρακτηριστικά του είναι ο έλεγχος σφαλμάτων, πεδία μεγέθους των τμημάτων κ.α.
- **Ταξινόμηση μηνυμάτων (sequencing)**
Ταξινόμηση με τη σωστή σειρά των τμημάτων όταν αυτά φτάσουν στον προορισμό τους. Εφαρμόζεται όταν τα διάφορα πακέτα μπορούν να ακολουθήσουν διαφορετικό δρόμο.
- **Ορισμός προτεραιότητας διεκπεραίωσης (priority)**
Ορισμένα μηνύματα είναι δυνατόν να έχουν προτεραιότητα έναντι κάποιων άλλων.
- **Διαδικασίες ασφάλειας (security)**
Εξασφαλίζει ότι το μήνυμα θα μεταδοθεί χωρίς κίνδυνο υποκλοπών, παρεμβολών και αλλοιώσεων.
- **Έλεγχος σφαλμάτων (error control)**
Αναγνώριση εσφαλμένης μετάδοσης μηνύματος.
- **Έλεγχος ροής (flow control)**
Ο αποδέκτης των μηνυμάτων μπορεί να ζητήσει περιορισμό προσωρινής ή μόνιμης διακοπής της αποστολής δεδομένων.
- **Διευθυνσιοδότηση μηνυμάτων (addressing)**
Απονομή κωδικών διευθύνσεων σε τερματικούς σταθμούς του δικτύου.

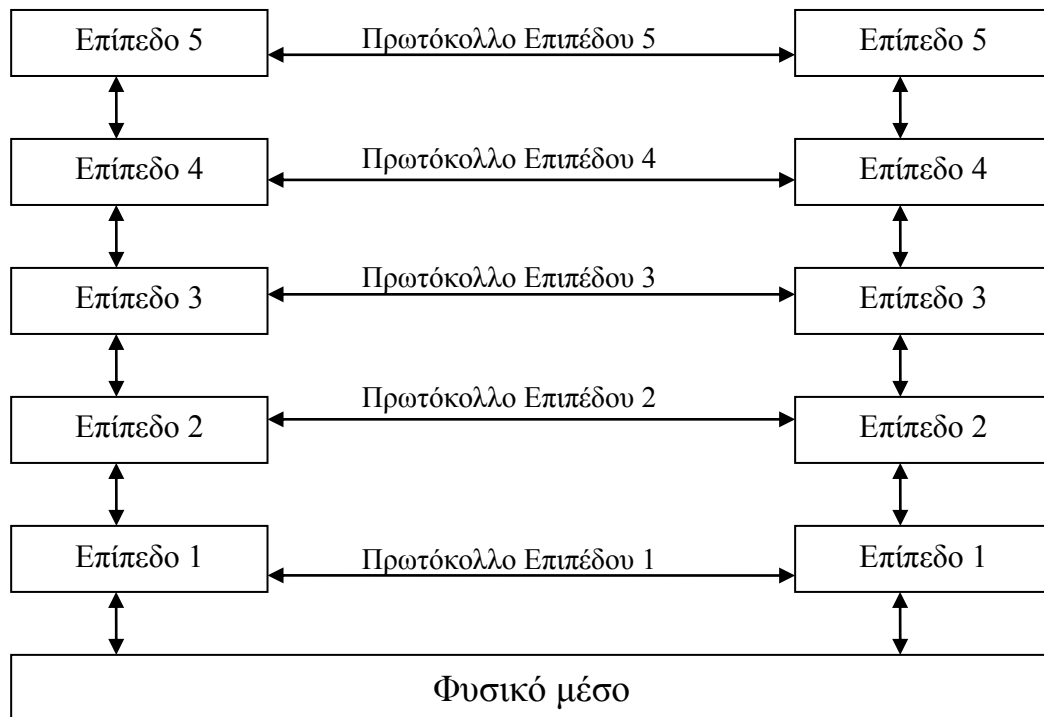
1.3. Ιεραρχία

Τα πρώτα δίκτυα υπολογιστών σχεδιάστηκαν κατά κύριο λόγο ως προς το υλικό και δευτερευόντως εξέταζαν το λογισμικό. Αυτή η στρατηγική πλέον δεν αποδίδει. Σήμερα το λογισμικό είναι δομημένο σε υψηλό βαθμό.

Για να μειωθεί η διαδραστική τους πολυπλοκότητα, τα περισσότερα δίκτυα οργανώνονται σε μία **στοίβα επιπέδων** (layers ή levels), τα οποία χτίζονται το ένα πάνω στο άλλο. Το πλήθος των επιπέδων, το όνομα, τα περιεχόμενα και η λειτουργία του κάθε επιπέδου διαφέρουν από δίκτυο σε δίκτυο. η δουλειά του κάθε

επιπέδου είναι να προσφέρει κάποιες υπηρεσίες στα ανώτερα επίπεδα κρύβοντας από τα επίπεδα αυτά τις λεπτομέρειες της υλοποίησης των παρεχόμενων υπηρεσιών. Με άλλα λόγια, κάθε επίπεδο είναι ένα είδος εικονικής μηχανής που παρέχει κάποιες υπηρεσίες στο ανώτερο επίπεδο.

Το επίπεδο n σε μία μηχανή πραγματοποιεί μία συνομιλία με το επίπεδο n σε κάποια άλλη μηχανή. Οι κανόνες και οι συμβάσεις που χρησιμοποιούνται σε μία τέτοια συνομιλία ονομάζονται συνολικά πρωτόκολλο του επιπέδου n .



Η παραπάνω εικόνα δείχνει ένα δίκτυο πέντε επιπέδων.

Στην πραγματικότητα κανένα στοιχείο δεν μεταδίδεται άμεσα από το επίπεδο n της μίας μηχανής στο επίπεδο n της άλλης. Αυτό που γίνεται είναι κάθε επίπεδο να μεταβιβάζει τα δεδομένα και τις πληροφορίες ελέγχου στο επίπεδο που βρίσκεται κάτω από αυτό μέχρι να φτάσουμε στο κατώτερο επίπεδο. Κάτω από το κατώτερο επίπεδο βρίσκεται το φυσικό μέσο, μέσω του οποίου γίνεται η πραγματική επικοινωνία.

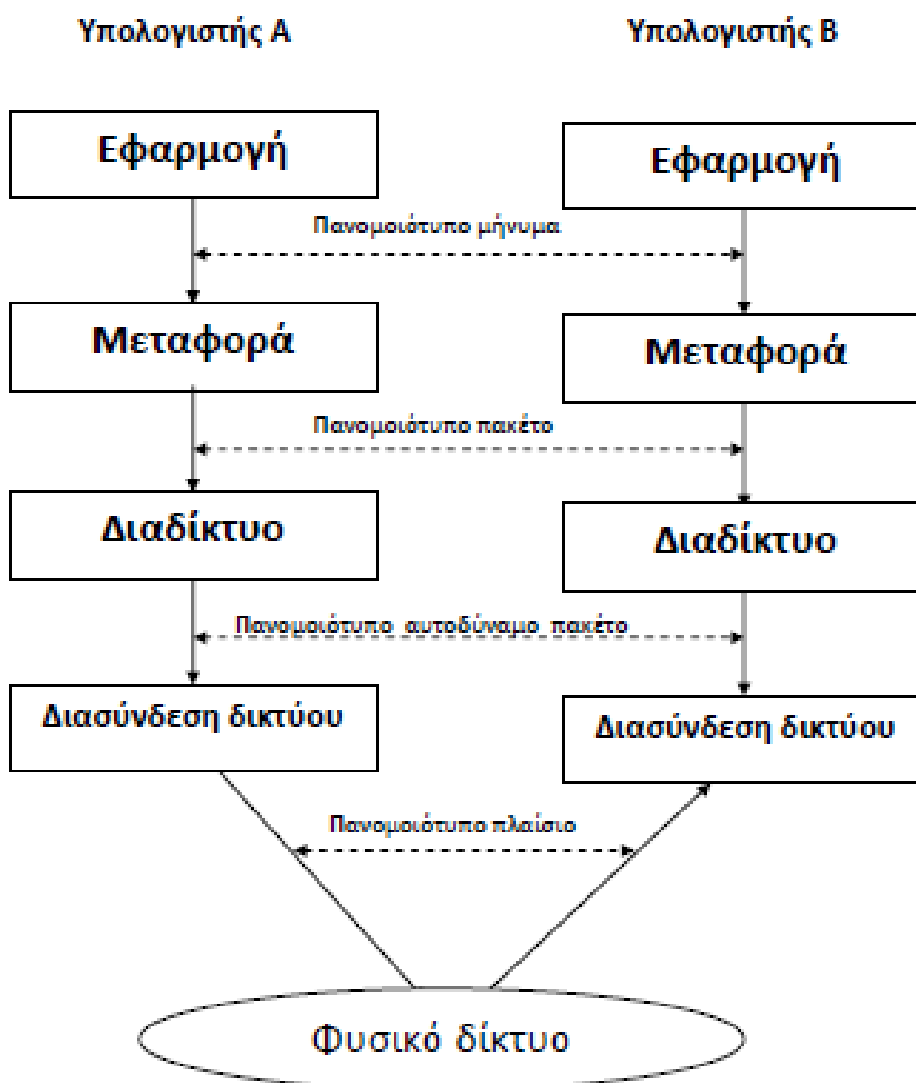
1.4. Αρχή της διαστρωμάτωσης πρωτοκόλλων

Ανεξάρτητα από τα αν χρησιμοποιούν κάποιο μοντέλο διαστρωμάτωσης ή τις λειτουργίες των επιπέδων, η λειτουργία των πολυεπίπεδων πρωτοκόλλων βασίζεται σε μια θεμελιώδη αρχή: την αρχή της διαστρωμάτωσης. Αυτή η αρχή λέει ότι:

Τα πολυεπίπεδα πρωτόκολλα είναι σχεδιασμένα έτσι ώστε το επίπεδο n του προορισμού να λαμβάνει το ίδιο ακριβώς αντικείμενο που στέλνει το επίπεδο n της προέλευσης.

Με αυτή την αρχή, ο σχεδιαστής πρωτοκόλλων μπορεί και επικεντρώνεται σε ένα επίπεδο κάθε φορά, χωρίς να τον απασχολεί η απόδοση των υπολοίπων. Για παράδειγμα, κατά τη δημιουργία μιας εφαρμογής για την μεταφορά αρχείων, ο σχεδιαστής ασχολείται μόνο με δύο αντίγραφα του προγράμματος εφαρμογής που εκτελείται σε δύο υπολογιστές και επικεντρώνεται στα μηνύματα που πρέπει να ανταλλάξουν οι υπολογιστές για να επιτευχθεί η μεταφορά των αρχείων. Ο σχεδιαστής υποθέτει ότι η εφαρμογή στον έναν υπολογιστή λαμβάνει τα ίδια ακριβώς δεδομένα που στέλνει η εφαρμογή του άλλου υπολογιστή.

Στο παρακάτω σχήμα φαίνεται πιο καθαρά πως λειτουργεί η αρχή της διαστρωμάτωσης:



2. Το μοντέλο αναφοράς ISO/OSI

2.1. Εισαγωγή

Το πρότυπο διασύνδεσης ανοικτών συστημάτων (OSI – Open Systems Interconnection-Reference Model) αναπτύχθηκε από τον Διεθνή Οργανισμό Τυποποίησης (ISO –International Standards Organization) το 1983 και ονομάστηκε έτσι γιατί αποτέλεσε τη βάση αναφορών και το πλαίσιο καθορισμού των προτύπων διασύνδεσης ανοικτών συστημάτων. Στόχος της ανάπτυξης αυτού του μοντέλου ήταν η δυνατότητα επικοινωνίας των συστημάτων που προέρχονταν από διαφορετικούς κατασκευαστές και η υποστήριξη εφαρμογών κατανεμημένης επεξεργασίας, ανεξάρτητα από το υλικό και το λογισμικό που χρησιμοποιείται.

Πρέπει να τονιστεί ότι το OSI είναι ένα μοντέλο και όχι ένα πρωτόκολλο που μπορεί να εγκατασταθεί και να τρέξει σε κάποιο σύστημα. Οπότε αυτό που κάνουν οι μηχανικοί είναι να αναπτύσσουν πρωτόκολλα που συμμορφώνονται στο μοντέλο OSI.

Το μοντέλο αναφοράς OSI αποτελείται από επτά ανεξάρτητα μεταξύ τους επίπεδα ή στρώματα (layers), καθένα από τα οποία υλοποιεί ένα συγκεκριμένο πρωτόκολλο. Ακολουθεί την αρχιτεκτονική των στρωμάτων ή επιπέδων, σύμφωνα με την οποία οι λειτουργίες του ανοικτού συστήματος στο οποίο αναφέρεται διαμοιράζονται σε ένα σύνολο κατακόρυφα διαρθρωμένων επιπέδων. Οι αρχές που εφαρμόστηκαν για να καταλήξουμε σε αυτά τα επτά επίπεδα είναι οι εξής:

- ✓ Ένα επίπεδο πρέπει να χρησιμοποιείται εκεί που χρειάζεται διαφορετικός αριθμός αφαίρεσης.
- ✓ Κάθε επίπεδο πρέπει να εκτελεί μία σαφώς καθορισμένη λειτουργία.
- ✓ Η λειτουργία κάθε επιπέδου πρέπει να επιλέγεται με βάση τα καθορισμένα διεθνώς πρωτόκολλα.
- ✓ Τα σύνορα των επιπέδων πρέπει να επιλέγονται έτσι ώστε να ελαχιστοποιείται η ροή πληροφοριών μέσω της διασύνδεσης των επιπέδων.
- ✓ Το πλήθος των επιπέδων πρέπει να είναι αρκετά μεγάλο έτσι ώστε να μην χρειάζεται να ανακατεύονται χωρίς λόγο διαφορετικές λειτουργίες στο ίδιο επίπεδο και ταυτόχρονα αρκετά μικρό ώστε η αρχιτεκτονική να μην είναι πολύπλοκη.

2.2. Επίπεδα μοντέλου OSI

Το σύνολο των επιπέδων που υλοποιούνται στο μοντέλο αναφοράς OSI, αρχίζοντας από το χαμηλότερο (επίπεδο 1) και προχωρώντας προς το υψηλότερο (επίπεδο 7) είναι το ακόλουθο:

✓ **Επίπεδο 1- φυσικό επίπεδο:**

Το φυσικό επίπεδο είναι υπεύθυνο για την μετάδοση των δυαδικών ψηφίων (bit) μέσω των διαφόρων φυσικών μέσων. Καθορίζει του τύπους καλωδίωσης που χρησιμοποιούνται (οπτική ίνα, ομοαξονικό, UTP), την τεχνική μετάδοσης (baseband ή broadband), είδη συνδετήρων (RJ-45, AUI, BNC, RS-232). Ακόμη, προσφέρει μεταφορά των δεδομένων σε μορφή bit, σύγχρονα ή ασύγχρονα και η επισήμανση σφαλμάτων μετάδοσης.

✓ **Επίπεδο 2-επίπεδο σύνδεσης δεδομένων:**

Είναι υπεύθυνο για την δημιουργία πακέτων προορισμού στα οποία επισυνάπτεται διεύθυνση προορισμού και αποστολές καθώς και για έλεγχο σφαλμάτων. Καθορίζει επίσης τον τρόπο πρόσβασης προς το μέσο μετάδοσης.

✓ **Επίπεδο 3- επίπεδο δικτύου**

Είναι υπεύθυνο για τις λειτουργίες δρομολόγησης και διευθυνσιοδότησης των υπολογιστών και την μετατροπή των λογικών διευθύνσεων σε φυσικές διευθύνσεις. Εφαρμόζει προτεραιότητα στην μετάδοση των δεδομένων, Routing, δημιουργία και ενοποίηση πακέτων.

✓ **Επίπεδο 4- επίπεδο μεταφοράς:**

Αναλαμβάνει τη μεταφορά δεδομένων απ' άκρη σ' άκρη στο δίκτυο, χρησιμοποιώντας τις υπηρεσίες των χαμηλότερων επιπέδων. Παρέχει βεβαίωση παραλαβής των πακέτων, αναμεταδίδει τα χαμένα πακέτα εφόσον απαιτείται, τηρεί τη σειρά των πακέτων και εφαρμόζει πολυπλεξία

✓ **Επίπεδο 5- επίπεδο συνόδου:**

Ελέγχει την έναρξη, την συντήρηση και τον τερματισμό των συνδέσεων του επιπέδου 4.

✓ **Επίπεδο 6- επίπεδο παρουσίασης:**

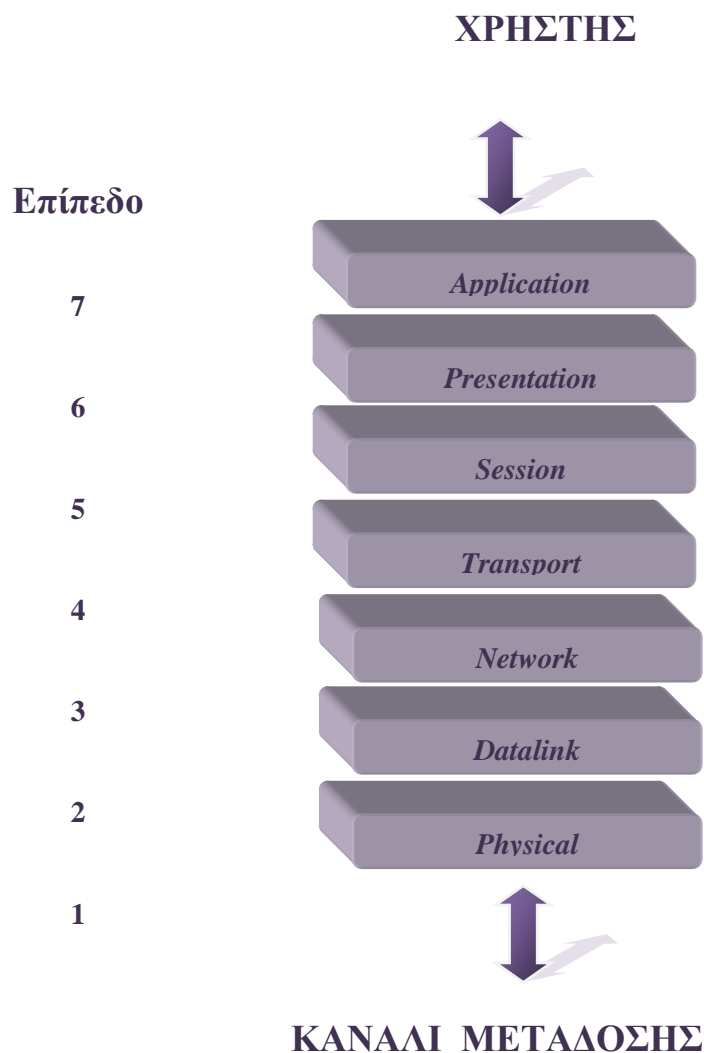
Αναλαμβάνει τη μορφοποίηση και την κωδικοποίηση των δεδομένων που ανταλλάσσονται.

✓ **Επίπεδο 7- επίπεδο εφαρμογών:**

Το επίπεδο εφαρμογών περιέχει μία ποικιλία πρωτοκόλλων που απαιτούνται από τους χρήστες. Ένα ευρέως χρησιμοποιούμενο πρωτόκολλο εφαρμογής είναι το Πρωτόκολλο Μεταφοράς Υπέρ-Κειμένου (Hypertext Transfer Protocol- HTTP), το οποίο είναι η βάση του Παγκόσμιου Ιστού.

Στο σχήμα που ακολουθεί βλέπουμε τα επτά επίπεδα του μοντέλου.

Τα επίπεδα 1,2,3 αφορούν τις υπηρεσίες-λειτουργίες που προσφέρονται από το δίκτυο, ενώ τα επίπεδα 4,5,6,7 έχουν να κάνουν με τις λειτουργίες της εφαρμογής του χρήστη. Επειδή, ακριβώς όπως αναφέραμε παραπάνω, κάθε επίπεδο εκτελεί μία μοναδική λειτουργία και ταυτόχρονα όλα μαζί μεταφέρουν πληροφορία προς τα πάνω, είναι εύκολο για τους προγραμματιστές και τους μηχανικούς δικτύων να λύνουν τα προβλήματα που προκύπτουν στο δίκτυο.



2.3. Πρωτόκολλα μεταφοράς (Transport Protocols)

Όπως είδαμε στο 2.2. το επίπεδο μεταφοράς του επιπέδου OSI διεκπεραιώνει τη μεταφορά των δεδομένων από χρήστη σε χρήστη, απαλλάσσοντας έτσι τα ανώτερα επίπεδα από κάθε φροντίδα να προσφέρουν αξιόπιστη μεταφορά δεδομένων από το ένα άκρο της επικοινωνίας στο άλλο.

Τα πρωτόκολλα μεταφοράς είναι υπεύθυνα για την επίτευξη μίας σύνδεσης και την εξασφάλιση ότι όλα τα δεδομένα θα φτάσουν στον προορισμό τους με ασφάλεια. Ορίζονται στο τέταρτο επίπεδο του μοντέλου OSI, το επίπεδο μεταφοράς. Συνήθως ο όρος «πρωτόκολλο μεταφοράς» υπονοεί υπηρεσίες μεταφοράς, οι οποίες περιλαμβάνουν τα δεδομένα που χρησιμοποιούνται στα χαμηλότερα επίπεδα, τα οποία μεταφέρονται σε μορφή πακέτων από τον έναν κόμβο στον άλλο.

Τα πρωτόκολλα μεταφοράς πολυμέσων δεν χρειάζεται να είναι πολύπλοκα ή «βαριές» εφαρμογές, όπως είναι ένα πρωτόκολλο για πολλές χρήσεις σαν το TCP/IP. Ο σκοπός των πρωτοκόλλων μεταφοράς είναι να προσφέρουν υπηρεσίες οι οποίες είναι συγκεκριμένες σε ένα λογικό εύρος εφαρμογών και να προσφέρουν υπηρεσίες οι οποίες είναι εξειδικευμένες και δεν είναι κοινές σε όλες τις εφαρμογές. Άρα αυτά τα πρωτόκολλα δεν είναι απαραίτητα στις γενικές υπηρεσίες δικτύων.

Με άλλα λόγια πρέπει να προσφέρουν υπηρεσίες για ένα κοινό σύνολο υπηρεσιών πολυμέσων, οι οποίες θα μπορούν εύκολα να διακριθούν σε σχέση με απλές υπηρεσίες μεταφοράς δεδομένων. Για παράδειγμα:

- Μία βασική υπηρεσία πλαισίου είναι απαραίτητη για να ορίζει την μονάδα μεταφοράς, τυπικό κοινό με την μονάδα συγχρονισμού.
- Μπορεί να είναι απαραίτητη η πολυπλεξία μιας και μπορεί να θέλουμε να δώσουμε διαφορετικά μέσα σε διαφορετικά κανάλια δεδομένων, αλλά μπορεί και να θέλουμε να προσκολλήσουμε όλων των ειδών τα μέσα στο ίδιο IP πακέτο.

Το συνηθέστερο παράδειγμα πρωτοκόλλου μεταφοράς είναι το TCP (Transmission Control Protocol). Άλλα πρωτόκολλα μεταφοράς είναι τα UDP (User Datagram Protocol), SCTP (Stream Control Transmission Protocol) κ.α.

2.4. Μετάδοση βέλτιστης προσπάθειας

Η μετάδοση βέλτιστης προσπάθειας (best effort delivery) περιγράφει μία υπηρεσία διαδικτύου κατά την οποία το διαδίκτυο δεν εγγυάται ότι τα δεδομένα θα μεταδοθούν ή ότι ένας χρήστης θα ενημερωθεί για την ποιότητα της υπηρεσίας που του προσφέρεται ή τις προτεραιότητες. Σε ένα δίκτυο βέλτιστης προσπάθειας όλοι οι χρήστες λαμβάνουν υπηρεσίας βέλτιστης προσπάθειας, γεγονός που σημαίνει ότι ο ρυθμός δεδομένων αποστολής ή λήψης όπως και ο χρόνος μετάδοσης εξαρτάται κάθε φορά από την τρέχουσα κίνηση του δικτύου.

Το UDP πρωτόκολλο προσφέρει την ίδια μετάδοση βέλτιστης προσπάθειας που προσφέρει και το IP πρωτόκολλο. Για αυτά θα μιλήσουμε παρακάτω.

2.5. Θύρες (ports)

Στις επόμενες ενότητες θα συναντήσουμε αρκετές φορές την έννοια της θύρας.

Στην επιστημή των υπολογιστών και πιο συγκεκριμένα στο διαδίκτυο υπολογιστών ο όρος «αριθμός θύρας» είναι μέρος της πληροφορίας διεύθυνσης που χρησιμοποιείται για να αναγνωριστούν οι αποστολείς και οι αποδέκτες των μηνυμάτων που στέλνονται. Τόσο το TCP όσο και το UDP χρησιμοποιούν ευρέως τους αριθμούς θυρών.

Οι αριθμοί θυρών συσχετίζονται με διευθύνσεις του διαδικτύου και δουλεύουν ακριβώς όπως οι επεκτάσεις του τηλεφώνου. Δηλαδή, ακριβώς όπως ένα βασικό τηλεφωνικό κέντρο μιας εταιρείας μπορεί να καλέσει κάθε εργαζόμενο με βάση έναν αριθμό –καλείται αριθμός επέκτασης- έτσι και ο υπολογιστής έχει μία βασική διεύθυνση και ένα σύνολο από αριθμούς θυρών που διαχειρίζονται τις εισερχόμενες και τις εξερχόμενες συνδέσεις.

Και στο TCP και στο UDP οι αριθμοί θυρών ξεκινούν από το 0 και φτάνουν το 65535.

3. Μοντέλο αναφοράς TCP/IP

3.1. Τι είναι το TCP/IP

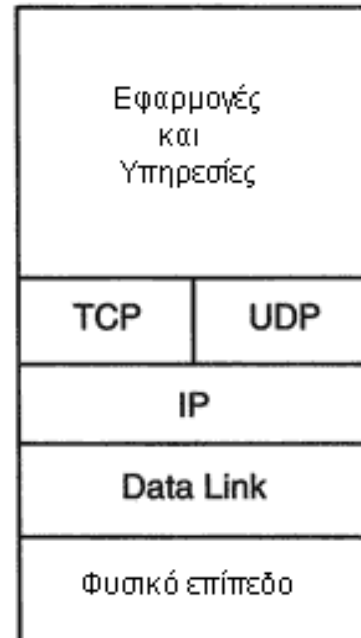
Η ικανότητα διασύνδεσης πολλών δικτύων με διαφανή τρόπο, η ικανότητα του δικτύου να επιβιώνει από απώλειες στο υλικό του υποδικτύου χωρίς να τερματίζονται οι υπάρχουσες συνδέσεις και η δυνατότητα χρήσης εφαρμογών με ριζικά διαφορετικές απαιτήσεις, από μεταφορά αρχείων έως μετάδοση ομιλίας σε πραγματικό χρόνο, οδήγησαν στη σχεδίαση μιας αρχιτεκτονικής γνωστή ως **Μοντέλο Αναφοράς TCP/IP**.

Το **TCP/IP (Transmission Control Protocol/ Internet Protocol, δηλαδή Πρωτόκολλο Ελέγχου Εκπομπής/ Πρωτόκολλο του Διαδικτύου)** είναι μία εναλλακτική πρόταση του μοντέλου OSI και αποτελείται από μία συλλογή πρωτοκόλλων επικοινωνίας στα οποία βασίζεται το Διαδίκτυο αλλά και μεγάλο ποσοστό των εμπορικών δικτύων. Τα αρχικά TCP/IP αναφέρονται σε δύο από τα σημαντικότερα πρωτόκολλα που χρησιμοποιούνται στο Διαδίκτυο, δηλαδή **στο TCP και στο IP** και όχι σε όλα τα πρωτόκολλα του Διαδικτύου.

Το TCP/IP μπορεί να θεωρηθεί ως ένα σύνολο τεσσάρων επιπέδων (layers) που το καθένα λύνει ένα σύνολο προβλημάτων που

εμφανίζονται στην μετάδοση δεδομένων και παρέχει μία καθορισμένη με σαφήνεια υπηρεσία στα πρωτόκολλα

ανώτερου επιπέδου. Τα ανώτερα επίπεδα είναι πιο κοντά στον άνθρωπο-χρήστη, ασχολούνται με αφηρημένα στοιχεία και στηρίζονται στα χαμηλότερα επίπεδα για να μετατρέψουν τα στοιχεία αυτά σε φυσικές μορφές που μπορούν τελικά να μεταδοθούν μέσα από τα δίκτυα.



Τα επίπεδα αυτά είναι:

- **Επίπεδο διαδικτύου**
Η δουλειά του είναι να επιτρέπει στους υπολογιστές υπηρεσίας να εισάγουν τα πακέτα τους σε οποιοδήποτε δίκτυο και αυτά να ταξιδεύουν ανεξάρτητα προς τον προορισμό τους.
- **Επίπεδο μεταφοράς**
Έχει σχεδιαστεί για να επιτρέπει στις ομότιμες οντότητες στους υπολογιστές υπηρεσίας να προέλευσης και προορισμού να συνομιλούν. Έχουν οριστεί δύο πρωτόκολλα μεταφοράς από άκρου εις άκρο. Το Πρωτόκολλο Ελέγχου Μετάδοσης ή TCP και το Πρωτόκολλο Αυτοδύναμων Πακέτων ή UDP.
- **Επίπεδο εφαρμογών**
Περιέχει όλα τα πρωτόκολλα ανώτερου επιπέδου.
- **Το επίπεδο διασύνδεσης μεταξύ υπολογιστή υπηρεσίας και δικτύου**
Το μοντέλο αναφοράς TCP/IP δε λέει και πολλά για το τι συμβαίνει εκεί, αλλά απλώς παρατηρεί ότι ο υπολογιστής υπηρεσίας πρέπει να συνδέεται με το δίκτυο χρησιμοποιώντας κάποιο πρωτόκολλο έτσι ώστε να μπορεί να στέλνει πακέτα IP.

Τα πρωτόκολλα που υπάρχουν στο TCP/IP για τον χειρισμό της επικοινωνίας των δεδομένων είναι:

- Το TCP/IP (Transmission Control Protocol), για επικοινωνία ανάμεσα σε εφαρμογές

- Το UDP (User Datagram Protocol), για απλή επικοινωνία ανάμεσα σε εφαρμογές
- Το IP (Internet Protocol), για επικοινωνία ανάμεσα σε υπολογιστές
- Το ICMP (Internet Control Message Protocol), για λάθη και στατιστικές
- Το DHCP (Dynamic Host Configuration Protocol), για δυναμική διευθυνσιοδότηση.

Πρέπει να τονίσουμε ότι αν και το TCP παρουσιάζεται ως μέρος της συλλογής πρωτοκόλλων TCP/IP, στην πραγματικότητα είναι ένα ανεξάρτητο πρωτόκολλο γενικής χρήσης που μπορεί να χρησιμοποιηθεί και με άλλα συστήματα παράδοσης. Για παράδειγμα είναι δυνατόν να χρησιμοποιηθεί σε ένα δίκτυο όπως το Ethernet αλλά και σε κάποιο περίπλοκο διαδίκτυο.

3.2. Internet Protocol - IP

Το **IP** θεωρείται από τα σημαντικότερα πρότυπα που έχουν καθιερωθεί στον χώρο των δικτύων επικοινωνίας υπολογιστών. Η λειτουργία του ταιριάζει απόλυτα σε περιβάλλοντα διασυνδεδεμένων ετερογενών δικτύων υπολογιστών που αποτελούνται από μηχανές ποικίλων μεγεθών και υπολογιστικής ισχύος.

Η πιο σημαντική Internet υπηρεσία αποτελείται από ένα:

- μη-αξιόπιστο, γιατί η μεταφορά πακέτου δεν είναι εγγυημένη
- καλύτερης προσπάθειας, γιατί το Internet κάνει μία αρκετά καλή προσπάθεια για να μεταφέρει τα πακέτα
- χωρίς συνδέσεις, γιατί κάθε πακέτο αντιμετωπίζεται ανεξάρτητα από τα άλλα

σύστημα διανομής πακέτων ανάλογη με την υπηρεσία που παρέχεται από το υλικό ενός δικτύου.

Το **IP πρωτόκολλο** ορίζει το μηχανισμό της μη-αξιόπιστης, χωρίς συνδέσεις μετάδοσης πακέτων. Ορίζει τη βασική μονάδα μεταφοράς των δεδομένων και την ακριβή μορφή των δεδομένων αυτών καθώς μεταδίδονται στο Διαδίκτυο. Επιπλέον του τυπικού προσδιορισμού της μορφής των δεδομένων, το IP περιλαμβάνει και ένα σύνολο κανόνων που ορίζουν τον τρόπο με τον οποίο θα πρέπει να επεξεργάζονται τα πακέτα και τον τρόπο διαχείρισης λαθών. Πιο συγκεκριμένα, το IP ενσωματώνει την ιδέα της μη-αξιόπιστης μεταφοράς και της σχετικής με αυτή ιδέα της δρομολόγησης πακέτων. Το IP βρίσκεται σε αντιστοιχία υπηρεσιών με το OSI layer 3- network layer.

3.3. Transmission Control Protocol - TCP

3.3.1. Εισαγωγή

Ένα επίπεδο πάνω πιο πάνω από το IP, στο επίπεδο μεταφοράς (**transport layer**), βρίσκονται τα **TCP και UDP**. Οι software βιβλιοθήκες των TCP, UDP χρησιμοποιούν το IP και παρέχουν ένα πιο απλοποιημένο περιβάλλον για να γραφούν δικτυακές εφαρμογές.

Το TCP σχεδιάστηκε με βασικό στόχο την υποστήριξη δυνατοτήτων αξιόπιστης επικοινωνίας μεταξύ ζευγαριών διαδικασιών σε υπολογιστές που βρίσκονται σε διαφορετικά αλλά διασυνδεδεμένα δίκτυα επικοινωνίας. Είναι το πρωτόκολλο που χρησιμοποιείται κατεξοχήν στο Διαδίκτυο. Το TCP δημιουργεί μία ατελείωτη σειρά από μηνύματα, τα οποία φαίνονται σαν μία συνεχής ροή δεδομένων. Αυτή η ροή είναι δύο κατευθύνσεων και ταυτόχρονα πολύ αξιόπιστη.

Το TCP είναι πρωτόκολλο επικοινωνίας, όχι λογισμικό.

3.3.2. Χαρακτηριστικά του TCP

Τα κυριότερα χαρακτηριστικά του TCP είναι:

- ✓ **Είναι connection protocol**
Αρα χρησιμοποιείται μόνο μεταξύ των δύο υπολογιστών. Πριν ξεκινήσει η μεταφορά δεδομένων πρέπει να γίνει ένα τριπλό handshaking (SYN, SYN-ACK, ACK) μεταξύ των δύο υπολογιστών και το αντίστοιχο στον τερματισμό της αποστολής.
- ✓ **Είναι αξιόπιστο**
Το TCP του παραλήπτη ενημερώνει συνεχώς το TCP του αποστολέα για το ποιο είναι το επόμενο πακέτο που περιμένει, σύμφωνα με τον αύξοντα αριθμό των πακέτων που έχει ήδη λάβει. Αν αντιληφθεί ότι κάποιο πακέτο χάθηκε στην πορεία τότε επιβάλλει επαναμετάδοση και αν το πακέτο δεν μπορεί να έρθει μετά από πολλαπλές επαναμεταδόσεις τότε η σύνδεση διακόπτεται.
- ✓ **Εγγυάται την σωστή σειρά άφιξης των δεδομένων στην εφαρμογή του παραλήπτη**
Όταν τα δεδομένα έρθουν στην είσοδο του παραλήπτη με λάθος σειρά τότε το TCP layer κρατάει αυτά τα δεδομένα μέχρι να έρθουν και τα προηγούμενά τους και στη συνέχεια τα παραδίδει στην εφαρμογή με την σωστή σειρά.

- ✓ **Απόρριψη διπλών δεδομένων**
- ✓ **Προσφέρει αυτοματοποιημένο έλεγχο ροής δεδομένων**
Όταν ο buffer του παραλήπτη γεμίσει τότε σταματάει προσωρινά την μετάδοση ή ελαττώνει τον ρυθμό μετάδοσής του μέχρι να αδειάσει ο buffer.
- ✓ **Προσφέρει αυτοματοποιημένο έλεγχο συμφόρησης**
Το TCP προσπαθεί να επιτύχει μία μέγιστη απόδοση μεταφοράς δεδομένων αποφεύγοντας την συμφόρηση δεδομένων στους routers του Διαδικτύου.
- ✓ **Εγγυάται την ακεραιότητα του «μονοπατιού επικοινωνίας»**

Εξαιτίας των παραπάνω χαρακτηριστικών του, το TCP χρησιμοποιείται και επιβάλλεται να χρησιμοποιείται όπου η ακεραιότητα των δεδομένων είναι υψίστης σημασίας. Τέτοιου είδους εφαρμογές είναι web surfing, e-mails, file transfers και οποιαδήποτε άλλη μεταφορά αρχείων δεδομένων ανάμεσα σε δύο υπολογιστές.

3.3.3. Κεφαλίδα του TCP

Το επόμενο σχήμα δείχνει την μορφή ενός τμήματος TCP. Κάθε τμήμα αρχίζει με μία κεφαλίδα σταθερής μορφής, μεγέθους 20 δυαδικών ψηφίων. Η σταθερή κεφαλίδα μπορεί να ακολουθείται από επιλογές κεφαλίδας, Μετά τις επιλογές, μπορεί να ακολουθούν μέχρι 65495 byte δεδομένων, όπου τα 20 πρώτα αναφέρονται στην κεφαλίδα IP και τα δεύτερα 20 στην κεφαλίδα TCP.

Θύρα προέλευσης				Θύρα προορισμού			
Αριθμός ακολουθίας							
Αριθμός επιβεβαίωσης							
Μήκος κεφαλίδας TCP	ΕΠΕ	ΕΠΙ	ΩΘΗ	ΕΠΑ	ΣΥΤ	ΤΕΡ	Μέγεθος παραθύρου
Επιλογές (0 η περισσότερες 32μπιτες λέξεις)							
Δεδομένα (προαιρετικά)							

- Θύρα προέλευσης και Θύρα προορισμού:
Καθορίζουν τα τοπικά τερματικά σημεία της σύνδεσης.
- Αριθμός ακολουθίας και αριθμός επιβεβαίωσης:
Το πακέτο που είναι στη σειρά και η επιβεβαίωσης ορθής λήψης ενός πακέτου.

- Μήκος κεφαλίδας TCP:
Πόσες 32μπιτες λέξεις περιέχει η κεφαλίδα TCP.
- ΕΠΕ: παίρνει την τιμή 1 αν είναι σε χρήση ο Δείκτης επειγόντων.
- ΕΠΙ: παίρνει την τιμή 1 για να δείξει ότι ο Αριθμός επιβεβαίωσης είναι έγκυρος.
- ΩΘΗ: υποδεικνύει δεδομένα στα οποία χρησιμοποιήθηκε η λειτουργία ΩΘΗΣΗ.
- ΕΠΑ: χρησιμοποιείται για την επαναφορά μιας σύνδεσης που έχει μπλεχτεί λόγω της κατάρρευσης ενός υπολογιστή υπηρεσίας ή εξαιτίας κάποιας άλλης αιτίας. Ακόμη χρησιμοποιείται για την απόρριψη ενός μη έγκυρου τμήματος ή για την άρνηση σε μία απόπειρα ανοίγματος σύνδεσης.
- ΣΥΓ: χρησιμοποιείται για την εγκαθίδρυση συνδέσεων.
- ΤΕΡ: χρησιμοποιείται για τον τερματισμό μιας σύνδεσης.
- Μέγεθος παραθύρου:
Δείχνει πόσα byte μπορούν να σταλούν ξεκινώντας από το byte που επιβεβαιώνεται.
- Άθροισμα ελέγχου:
Περιλαμβάνει την κεφαλίδα, τα δεδομένα και την εικονική ψευδοκεφαλίδα.
- Επιλογές:
Παρέχει μία μέθοδο προσθήκης βοηθητικών λειτουργιών που δεν καλύπτονται από την κανονική κεφαλίδα.

4. Πρωτόκολλο Αυτοδύναμων Πακέτων – UDP

4.1. Αναγνώριση του απόλυτου προορισμού

Τα λειτουργικά συστήματα στους περισσότερους υπολογιστές υποστηρίζουν **πολυπρογραμματισμό**, που σημαίνει ότι επιτρέπουν σε πολλαπλές εφαρμογές να εκτελούνται ταυτόχρονα. Αναφερόμαστε σε κάθε πρόγραμμα που εκτελείται με τους όρους διεργασία (process), task, πρόγραμμα εφαρμογών ή διεργασία επιπέδου

χρήστη και τα συστήματα ονομάζονται πολυεπεξεργαστικά συστήματα. Ο καθορισμός ότι μια συγκεκριμένη διεργασία σε μια συγκεκριμένη μηχανή είναι ο απόλυτος προορισμός για ένα datagram είναι κατά κάποιο τρόπο παραπλανητικός. Αυτό συμβαίνει για τρεις λόγους:

- Οι διεργασίες δημιουργούνται και καταστρέφονται δυναμικά με αποτέλεσμα οι αποστολές σπάνια να γνωρίζουν αρκετά ώστε να προσδιορίσουν μια διεργασία σε μια άλλη μηχανή.
- Θα θέλαμε να μπορούμε να αντικαθιστούμε τις διεργασίες που λαμβάνουν datagrams χωρίς να ενημερώνουμε όλους τους αποστολές.
- Χρειάζεται να αναγνωρίζουμε προορισμούς με βάση τις λειτουργίες που υλοποιούν χωρίς να γνωρίζουμε τις διεργασίες που υλοποιούν την λειτουργία.

Αντί να σκεφτόμαστε μια διεργασία ως την απόλυτη διεύθυνση, θα φανταστούμε ότι κάθε μηχανή περιέχει ένα σύνολο από αφηρημένα σημεία προορισμού που ονομάζονται **θύρες πρωτοκόλλου** (protocol ports). Κάθε θύρα πρωτοκόλλου αναγνωρίζεται από έναν θετικό ακέραιο. Το τοπικό λειτουργικό σύστημα παρέχει έναν μηχανισμό interface που χρησιμοποιούν οι διεργασίες για να καθορίσουν μια θύρα ή για να προσπελάσουν.

Τα περισσότερα λειτουργικά συστήματα παρέχουν σύγχρονη πρόσβαση στις θύρες. Από την πλευρά μιας συγκεκριμένης διεργασίας, η σύγχρονη πρόσβαση σημαίνει ότι οι υπολογισμοί σταματούν κατά τη διάρκεια μιας λειτουργίας πρόσβασης σε θύρα. Γενικά οι θύρες έχουν την δυνατότητα αποθήκευσης σε buffers, έτσι ώστε τα δεδομένα που φτάνουν προτού να είναι έτοιμη η διεργασία να τα δεχτεί να μην χαθούν. Για την αποθήκευση σε buffers, το λογισμικό του πρωτοκόλλου που βρίσκεται μέσα στα λειτουργικά συστήματα, τοποθετεί τα πακέτα που φτάνουν για κάποια συγκεκριμένη θύρα σε μια ουρά μέχρι μια διεργασία να τα εξάγει.

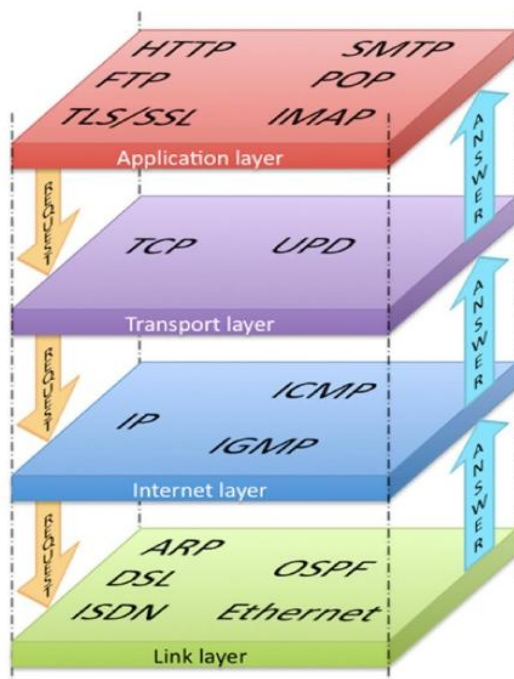
Για την επικοινωνία σε μια ξένη θύρα, ένας αποστολέας πρέπει να γνωρίζει και την IP διεύθυνση της μηχανής προορισμού και τον αριθμό πρωτοκόλλου της θύρας προορισμού σε αυτή τη μηχανή. Κάθε μήνυμα μεταφέρει και τον αριθμό της θύρας προορισμού στην ξένη μηχανή στην οποία στέλνεται το μήνυμα καθώς και ο αριθμός της θύρας της πηγής στη μηχανή στην οποία θα πρέπει να στέλνονται οι απαντήσεις. Έτσι είναι δυνατόν για οποιαδήποτε διεργασία που λαμβάνει ένα μήνυμα να απαντήσει στον αποστολέα.

4.2. Τι είναι το UDP

Στη συλλογή πρωτοκόλλων TCP/IP, το **Πρωτόκολλο Αυτοδύναμων Πακέτων Χρήστη (User Datagram Protocol - UDP)** παρέχει το βασικό μηχανισμό που χρησιμοποιούν τα προγράμματα εφαρμογών για να στέλνουν αυτοδύναμα πακέτα σε άλλα προγράμματα εφαρμογών. Το UDP παρέχει θύρες πρωτοκόλλων που χρησιμεύουν στη διάκριση μεταξύ των προγραμμάτων τα οποία εκτελούνται σε μια μηχανή. Δηλαδή, κάθε μήνυμα UDP περιλαμβάνει, εκτός από τα δεδομένα, έναν αριθμό θύρας προορισμού και έναν αριθμό θύρας προέλευσης, γεγονός που δίνει τη δυνατότητα στο λογισμικό UDP του προορισμού να παραδώσει το μήνυμα στο σωστό παραλήπτη και ακόμη επιτρέπει στον παραλήπτη να στείλει μια απάντηση.

Το UDP χρησιμοποιεί το Πρωτόκολλο Internet για να μεταφέρει ένα μήνυμα από μια μηχανή σε μια άλλη και παρέχει την ίδια μη αξιόπιστη, ασυνδεσμική παράδοση αυτοδύναμων πακέτων όπως και το IP. Δε χρησιμοποιεί σήματα επιβεβαίωσης για να σιγουρευτεί ότι τα μηνύματα έφτασαν στον προορισμό τους, δεν ταξινομεί τα εισερχόμενα μηνύματα και δεν παρέχει ανατροφοδότηση για να ελέγξει το ρυθμό ροής των πληροφοριών μεταξύ των μηνυμάτων. Άρα, τα μηνύματα UDP μπορεί να χαθούν, να αναπαραχθούν ή να φτάσουν στον προορισμό τους εκτός σειράς. Επιπλέον τα πακέτα μπορεί να φτάνουν πολύ γρήγορα και ο δείκτης να μην προλαβαίνει να τα επεξεργαστεί.

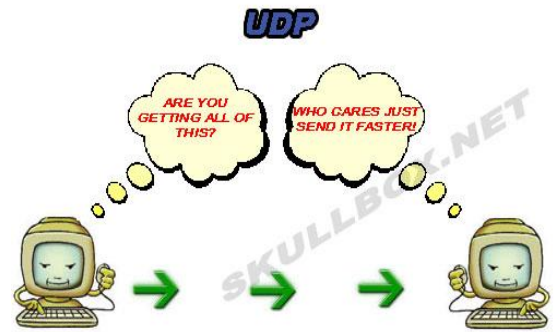
Ας σημειώσουμε ότι δεν είναι πολλοί εκείνοι που είναι εξοικειωμένοι με την χρήση του πρωτοκόλλου UDP έναντι του TCP και αυτό διότι κάθε κομμάτι πληροφορίας που μεταφέρεται, από οποιοδήποτε επίπεδο (εφαρμογής, συνόδου κ.α.) το TCP τους δίνει αξιοπιστία μεταφοράς.



4.3. Χαρακτηριστικά

Το UDP είναι ένα σχεδόν μηδενικό πρωτόκολλο, με την έννοια ότι κάνει τα ελάχιστα που μπορεί να κάνει ένα πρωτόκολλο επιπέδου μεταφοράς. **Οι μόνες υπηρεσίες που παρέχει είναι το checksum των προς μετάδοση δεδομένων και της πολυπλεξίας των ports επικοινωνίας του υπολογιστή.** Για αυτό το λόγο και πολλές φορές αναφέρεται με το όνομα **Unreliable Datagram Protocol**. Είναι, όμως και ένα από τα βασικά πρωτόκολλα που

χρησιμοποιούνται στο Διαδίκτυο. Μία ακόμη εναλλακτική ονομασία του πρωτοκόλλου είναι Universal Datagram Protocol.



Χρησιμοποιώντας το UDP τα προγράμματα μπορούν να στείλουν μικρά μηνύματα, γνωστά ως datagrams, το ένα στο άλλο. Τα χαρακτηριστικά του UDP είναι τα εξής:

- ✓ **Είναι αναξιόπιστο:**
Δεν μπορεί να εγγυηθεί τη ακεραιότητα ή τη σωστή σειρά άφιξης των δεδομένων. Τα πακέτα (datagrams) μπορούν να φτάσουν σε διαφορετική σειρά, να εμφανίζονται διπλά ή να μην έρθουν και καθόλου χωρίς καμία ειδοποίηση.
- ✓ **Είναι γρήγορο:**
Το παραπάνω χαρακτηριστικό του εξασφαλίζει μικρή καθυστέρηση (delay) και έτσι θεωρείται γρήγορο.
- ✓ **Πολλαπλή χρηστικότητα:**
Μπορεί να χρησιμοποιηθεί τόσο σε Unicast όσο και σε Multicast δίκτυα, καθώς δεν είναι connection protocol.
- ✓ **Είναι «ελαφρύ»:**
Το UDP δεν είναι απαιτητικό σε πόρους. Δεν δημιουργεί μεγάλο overhead στο δίκτυο, καθώς δεν ελέγχει αν όντως κάποιο πακέτο έχει φτάσει ή όχι.
- ✓ **Έχει μικρό header:**
Το UDP έχει 8 bytes header. Αυτό σημαίνει μικρότερο επιπλέον overhead στο δίκτυο.

4.4. Γιατί υπάρχει το UDP

Σύμφωνα με τα όσα αναφέραμε παραπάνω το TCP φαντάζει σαν μια ιδανική λύση για την σωστή και αξιόπιστη κυρίως μεταφορά δεδομένων. **Γιατί, όμως, υπάρχει το UDP ή ακόμα καλύτερα γιατί το χρησιμοποιούμε;**

Το UDP υπάρχει γιατί υπάρχουν εφαρμογές στις οποίες δεν μας ενδιαφέρει τόσο η ακεραιότητα των δεδομένων όσο τα δεδομένα να φτάσουν όσο πιο γρήγορα γίνεται στον παραλήπτη, έστω και με κάποια απώλεια. Εκεί δηλαδή που το TCP είναι αργό και δεν μας εξυπηρετεί, έρχεται να πάρει την θέση του το UDP.

Μερικές εφαρμογές που χρησιμοποιούν το UDP είναι οι παρακάτω:

- Εφαρμογές οι οποίες μεταδίδουν real-time audio/video, όπως IPVT, VoIP. Εδώ μας ενδιαφέρει τα δεδομένα να φτάνουν την σωστή χρονική στιγμή. Οποιαδήποτε απώλειά τους μας επηρεάζει μόνο στην ποιότητα του αναπαραγόμενου σήματος.
- Servers, οι οποίοι απαντάνε σε μικρά αιτήματα ενός τεράστιου αριθμού από clients, όπως στα online παιχνίδια. Οι servers δεν απασχολούνται με το να ελέγχουν την κατάσταση του κάθε connection και των παραμέτρων του, χρησιμοποιώντας UDP και έτσι μπορούν να εξυπηρετήσουν ένα πολύ μεγαλύτερο αριθμό χρηστών σε αντίθεση με το αν χρησιμοποιούσαν TCP.
- Κάποιες πολύ σημαντικές εφαρμογές όπως το Domain Name System (DNS), Simple Network Management Protocol (SNMP), Dynamic Host Configuration Protocol (DHCP), Routing Information Protocol (RIP).

Μετά από τα παραπάνω βλέπουμε ότι ένα πρόγραμμα που χρησιμοποιεί το πρωτόκολλο UDP πρέπει να ασχοληθεί με τα προβλήματα επικοινωνίας που μπορεί να προκύψουν:

την αξιόπιστη παράδοση, το packetization και την επανασυναρμολόγηση, τον έλεγχο ροής, την αποφυγή συμφόρησης κ.α. Επίσης, δεδομένου ότι το UDP στερείται μηχανισμών αποφυγής και ελέγχου δικτυακής συμφόρησης απαιτούνται network-based μηχανισμοί για να ελαχιστοποιηθούν τα πιθανά προβλήματα κατάρρευσης δικτύου λόγω ανεξέλεγκτα υψηλών ρυθμών αποστολής πακέτων UDP.

Με άλλα λόγια, δεδομένου ότι οι αποστολές UDP δεν μπορούν να ανιχνεύσουν συμφόρηση, τα στοιχεία των δικτύων, όπως οι routers, πρέπει να χρησιμοποιούν τεχνικές packet queuing και απόρριψης πακέτων για να ελέγχουν την υπερβολική κίνηση πακέτων UDP στα δίκτυα.

Ένα πρόγραμμα εφαρμογής που χρησιμοποιεί UDP είναι αποκλειστικά υπεύθυνο για τον χειρισμό του προβλήματος της αξιοπιστίας και όσων αυτό συνεπάγεται,

όπως η απώλεια μηνύματος, η αναπαραγωγή μηνύματος, η καθυστέρηση, η παράδοση εκτός σειράς και η απώλεια συνδετικότητας. Δυστυχώς, οι σχεδιαστές των προγραμμάτων εφαρμογών συχνά παραβλέπουν τα προβλήματα αυτά κατά τον σχεδιασμό του λογισμικού. Επιπλέον, οι προγραμματιστές συχνά δοκιμάζουν την απόδοση του λογισμικού δικτύων σε αξιόπιστα τοπικά δίκτυα με μικρή καθυστέρηση και για το λόγο αυτό ο έλεγχος μπορεί να μη δείξει τα πιθανά προβλήματα. Έτσι, πολλά προγράμματα εφαρμογών που βασίζονται στο UDP λειτουργούν καλά σε τοπικό περιβάλλον, παρουσιάζουν όμως τραγικά προβλήματα όταν χρησιμοποιούν σε μεγαλύτερα διαδίκτυα TCP.

5. Λειτουργικές προδιαγραφές UDP

5.1. Μορφή μηνύματος UDP

Κάθε μήνυμα UDP ονομάζεται αυτοδύναμο πακέτο χρήστη. Από την πλευρά του σχεδιασμού το αυτοδύναμο πακέτο χρήστη αποτελείται από δύο μέρη:

- i. μία κεφαλίδα UDP – UDP header και
- ii. μία περιοχή δεδομένων UDP- UDP data area

Στο παρακάτω σχήμα φαίνεται αναλυτικά η μορφή των πεδίων σε ένα αυτοδύναμο πακέτο UDP.

0	16	31
Θύρα πηγής UDP	Θύρα προορισμού UDP	
Μήκος μηνύματος UDP	Άθροισμα ελέγχου UDP	
Δεδομένα		
...		

Παρατηρούμε ότι η δομή του UDP πακέτου είναι πιο απλή από την δομή του TCP πακέτου, γεγονός που είναι λογικό καθώς όπως είπαμε το UDP είναι πιο απλό από το TCP.

Όπως φαίνεται στην εικόνα, η κεφαλίδα χωρίζεται σε τέσσερα πεδία 16 δυαδικών ψηφίων, τα οποία προσδιορίζουν την θύρα από την οποία στάλθηκε το μήνυμα, τη

θύρα για την οποία προορίζεται το μήνυμα, το μήκος του μηνύματος και ένα άθροισμα ελέγχου UDP.

Το πεδίο **Θύρα προέλευσης UDP** είναι η θύρα του αποστολέα από την οποία προήλθε το πακέτο. Περιέχει δηλαδή τον 16μπιτο αριθμό θύρας που χρησιμοποιείται για την αναγνώριση του σημείου εκκίνησης του πακέτου στο μηχάνημα της πηγής. Εάν ο παραλήπτης επιθυμεί απαντάει σε περίπτωση που θελήσει σε αυτή τη θύρα. Το συγκεκριμένο πεδίο δεν είναι υποχρεωτικό. Όταν χρησιμοποιείται προσδιορίζει τη θύρα στην οποία πρέπει να σταλούν οι απαντήσεις, ενώ στις περιπτώσεις που δεν χρησιμοποιείται θα πρέπει να έχει την τιμή μηδέν.

Αντίστοιχα και η **Θύρα προορισμού UDP** περιέχει τον 16μπιτο αριθμό θύρας πρωτοκόλλου UDP που χρησιμοποιείται για την αναγνώριση του τερματικού σημείου στο μηχάνημα του προορισμού.

Το πεδίο **Μήκος μηνύματος UDP** έχει μέγεθος 16 δυαδικά ψηφία και περιλαμβάνει το μέγεθος του πακέτου σε bytes. Το μικρότερο δυνατό μέγεθος είναι 8 bytes, αφού η κεφαλίδα αυτή καθ' αυτή καταλαμβάνει τόσο χώρο. Θεωρητικά το μέγεθος του UDP πακέτου δεν μπορεί να ξεπερνάει τα 65527 bytes, αλλά πρακτικά το όριο μειώνεται στα 65507 bytes λόγω διάφορων περιορισμών εισαγόμενων από το πρωτόκολλο IPv4 στο επίπεδο δικτύου.

Το πεδίο **Άθροισμα ελέγχου UDP** έχει μέγεθος 16 δυαδικά ψηφία και χρησιμοποιείται για την επαλήθευση της ορθότητας του πακέτου στο σύνολό του, δηλαδή τόσο της κεφαλίδας όσο και των δεδομένων. Στη συνέχεια το UDP πακέτο περνάει στο επίπεδο δικτύου, το οποίο αναλαμβάνει τη μετάδοσή του στο δίκτυο υπολογιστών. Το επίπεδο αυτό τοποθετεί μία ακόμη κεφαλίδα στο πακέτο, η οποία διαφέρει ανάλογα με την έκδοσή του. Το άθροισμα ελέγχου UDP είναι προαιρετικό. Αν το πεδίο Άθροισμα ελέγχου UDP έχει μηδενική τιμή, αυτό σημαίνει ότι το άθροισμα ελέγχου δεν έχει υπολογιστεί. Οι σχεδιαστές επέλεξαν αυτό να είναι προαιρετικό για να δώσουν τη δυνατότητα στις εφαρμογές να λειτουργούν με λίγη υπολογιστική υπερφόρτωση όταν χρησιμοποιούν UDP σε τοπικά δίκτυα υψηλής αξιοπιστίας. Μάλιστα, καθώς το IP δεν υπολογίζει το άθροισμα ελέγχου στο τμήμα δεδομένων του αυτοδύναμου πακέτου IP, το άθροισμα ελέγχου UDP παρέχει το μοναδικό τρόπο εγγύησης ότι τα δεδομένα έχουν φτάσει ανέπαφα στον προορισμό τους και θα πρέπει να χρησιμοποιείται.

5.2. Ψευδοκεφαλίδα UDP

Το άθροισμα ελέγχου καλύπτει περισσότερες πληροφορίες από όσες υπάρχουν στο ίδιο αυτοδύναμο πακέτο UDP. Προκειμένου το UDP να υπολογίσει το άθροισμα ελέγχου, τοποθετεί μία ψευδοκεφαλίδα (pseudo-header) στο αυτοδύναμο πακέτο, προσαρτά μια οκτάδα μηδενικών για να δώσει στο πακέτο μέγεθος πολλαπλάσιο

των 16 δυαδικών ψηφίων και υπολογίζει το άθροισμα ελέγχου για όλο το αντικείμενο. Η ψευδοκεφαλίδα μαζί με την οκτάδα που προστίθεται επιπλέον δεν μεταδίδονται με το αυτοδύναμο πακέτο UDP, άρα δεν συμπεριλαμβάνονται και στο μήκος. Προκειμένου να υπολογιστεί το άθροισμα ελέγχου, το λογισμικό πρώτα αποθηκεύει την τιμή μηδέν στο πεδίο Άθροισμα ελέγχου UDP και μετά συγκεντρώνει ένα συμπλήρωμα ως προς ένα από όλο το αντικείμενο, μαζί με την ψευδοκεφαλίδα, την κεφαλίδα UDP και τα δεδομένα του χρήστη.

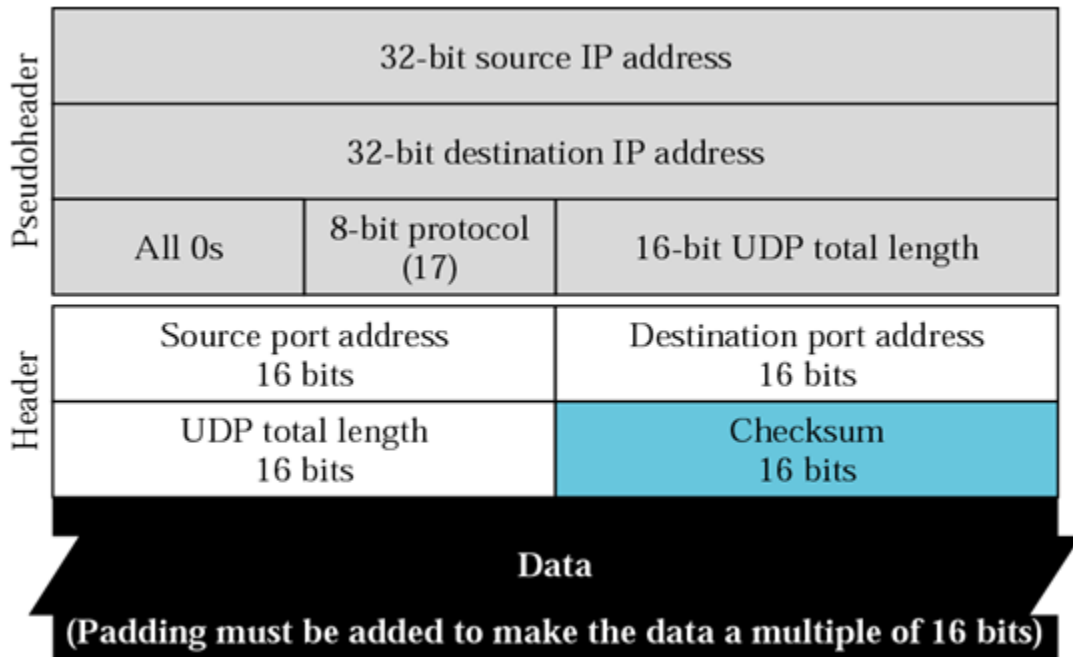
Η ψευδοκεφαλίδα χρησιμοποιείται για να επιβεβαιώσει ότι το αυτοδύναμο πακέτο UDP έχει φτάσει στο σωστό προορισμό. Να πούμε ότι, ο σωστός προορισμός αποτελείται από μια συγκεκριμένη μηχανή και μια συγκεκριμένη θύρα πρωτοκόλλου σε αυτή τη μηχανή. Η ίδια η κεφαλίδα UDP προσδιορίζει μόνο τον αριθμό της θύρας πρωτοκόλλου. Οπότε για να προσδιοριστεί ο προορισμός υπολογίζεται ένα άθροισμα ελέγχου στην μηχανή προέλευσης που καλύπτει τη διεύθυνση προορισμού IP καθώς επίσης και το αυτοδύναμο πακέτο UDP. Στον τελικό προορισμό, το λογισμικό UDP επαληθεύει το άθροισμα ελέγχου που έλαβε από την κεφαλίδα. Αν τα αθροίσματα ελέγχου συμφωνούν, τότε το αυτοδύναμο πακέτο πρέπει να φτάσει στον επιθυμητό υπολογιστή υπηρεσίας και στη σωστή θύρα πρωτοκόλλου του υπολογιστή αυτού.

Στην εικόνα που ακολουθεί φαίνονται οι δώδεκα οκτάδες της ψευδοκεφαλίδας που χρησιμοποιείται για τον υπολογισμό του αθροίσματος ελέγχου UDP.

0	8	16	31
Διεύθυνση πηγής IP			
Διεύθυνση προορισμού IP			
Μηδέν	Πρωτόκολλο	Μήκος UDP	

Τα πεδία Διεύθυνση προέλευσης IP και Διεύθυνση προορισμού IP περιέχουν τις διευθύνσεις IP προέλευσης και προορισμού και χρησιμοποιούνται για την αποστολή του μηνύματος UDP. Το πεδίο Πρωτόκολλο περιέχει τον κωδικό τύπου πρωτοκόλλου IP – για το UDP είναι 17- και το πεδίο Μήκος UDP περιέχει το μήκος του αυτοδύναμου πακέτου UDP, χωρίς την ψευδοκεφαλίδα.

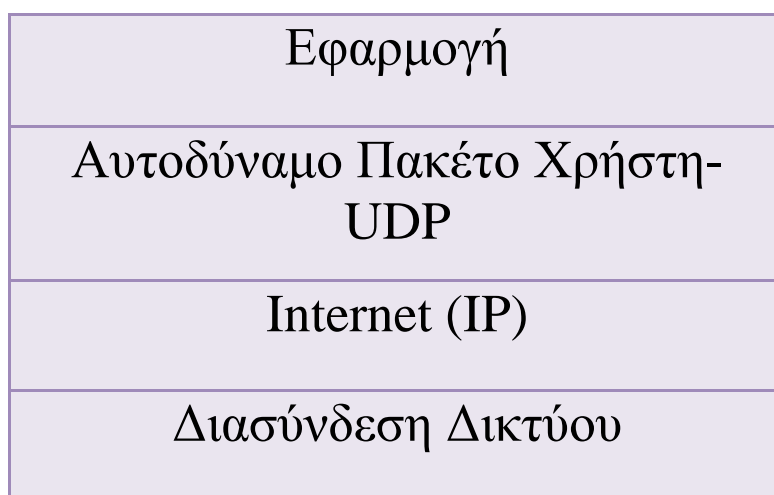
Στην εικόνα παρακάτω βλέπουμε την ψευδοκεφαλίδα για τον υπολογισμό του checksum.



5.3. Ενθυλάκωση και διαστρωμάτωση του UDP

Το UDP είναι πρωτόκολλο του επιπέδου μεταφοράς. Στο μοντέλο διαστρωμάτωσης πρωτοκόλλων το UDP βρίσκεται πάνω από το επίπεδο Πρωτοκόλλου Διαδικτύου (IP). Τα προγράμματα εφαρμογών προσπελούν το UDP, το οποίο χρησιμοποιεί IP για να στείλει και να λάβει αυτοδύναμα πακέτα. Το γεγονός ότι το UDP βρίσκεται πάνω από το IP σημαίνει ότι ένα πλήρες μήνυμα UDP ενθυλακώνεται σε ένα αυτοδύναμο πακέτο IP καθώς ταξιδεύει σε ένα διαδίκτυο.

Στη εικόνα που ακολουθεί βλέπουμε την διαστρωμάτωση του UDP ανάμεσα στα προγράμματα εφαρμογών και το IP.



Η ενθυλάκωση σημαίνει ότι το UDP τοποθετεί μία κεφαλίδα στα δεδομένα που στέλνει ο χρήστης και τα μεταβιβάζει στο IP. Με τη σειρά του το επίπεδο IP τοποθετεί μία κεφαλίδα σε αυτά που λαμβάνει από το UDP. Τέλος, το επίπεδο διασύνδεσης δικτύου ενθυλακώνει το πακέτο σε ένα πλαίσιο πριν το στείλει από τη μία μηχανή στην άλλη.

Τώρα θα δούμε τι γίνεται κατά την άφιξη του πακέτου. Όταν το πακέτο φτάνει στο χαμηλότερο επίπεδο του λογισμικού δικτύου αρχίζει να στέλνεται σταδιακά στα ανώτερα επίπεδα. Κάθε επίπεδο αφαιρεί μία κεφαλίδα πριν το μεταβιβάσει στο επόμενο επίπεδο και έτσι όταν το ανώτατο επίπεδο μεταβιβάσει το πακέτο στην λαμβάνουσα διεργασία έχουν αφαιρεθεί όλες οι κεφαλίδες.

Κατά τη διάρκεια της παραπάνω διαδικασίας πρέπει να λαμβάνουμε υπόψη μας την αρχή της διαστρωμάτωσης, η οποία εφαρμόζεται στο UDP και έτσι το αυτοδύναμο πακέτο UDP που λαμβάνεται από το IP στη μηχανή προορισμού είναι όμοιο με το αυτοδύναμο πακέτο που το UDP μεταβίβασε στο IP, στη μηχανή προέλευσης. Άρα και τα δεδομένα που μεταφέρει το UDP σε μια διεργασία χρήστη στη λαμβάνουσα μηχανή είναι ακριβώς ίδια με εκείνα που η διεργασία χρήστη μεταβίβασε στο UDP στη μηχανή προέλευσης.

Συνοψίζοντας, έχουμε να πούμε ότι το επίπεδο IP είναι υπεύθυνο μόνο για την μεταφορά δεδομένων μεταξύ δύο υπολογιστών υπηρεσίας σε ένα διαδίκτυο, και άρα μόνο η κεφαλίδα IP προσδιορίζει τους υπολογιστές προέλευσης και προορισμού. Από την άλλη το επίπεδο UDP είναι υπεύθυνο μόνο για την διαφοροποίηση μεταξύ προελεύσεων και προορισμών σε έναν υπολογιστή, δηλαδή προσδιορίζει τις θύρες προέλευσης και προορισμού σε έναν υπολογιστή υπηρεσίας.

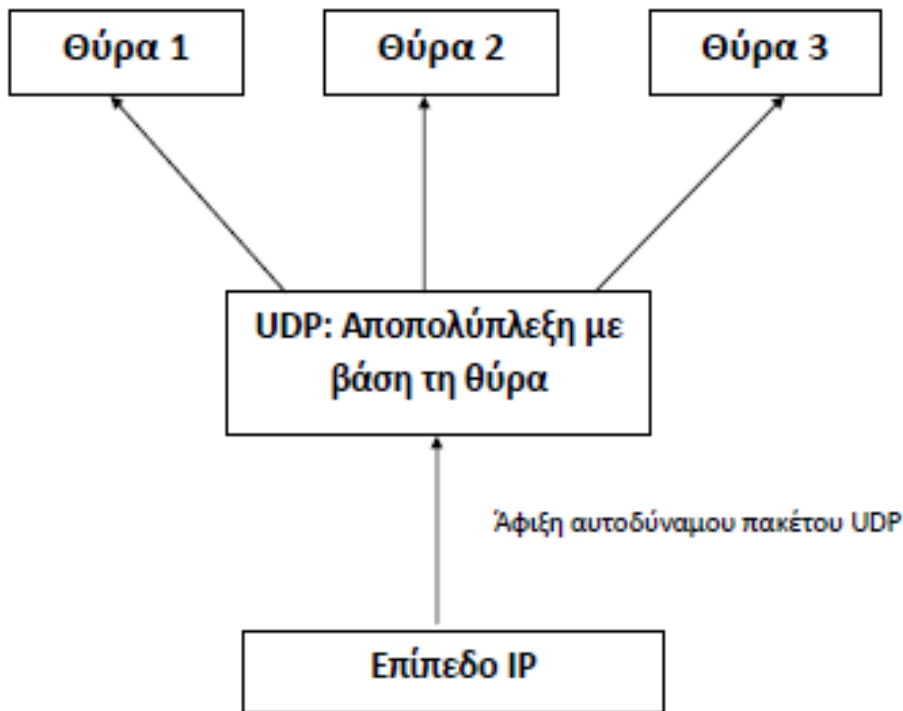
5.4. Πολύπλεξη και αποπολύπλεξη του UDP

Το λογισμικό σε όλα τα επίπεδα της ιεραρχίας πρωτοκόλλων θα πρέπει να πολυπλέκει ή να αποπολυπλέκει διάφορα αντικείμενα του επόμενου επιπέδου. Το λογισμικό UDP παρέχει ακόμα ένα παράδειγμα πολύπλεξης και αποπολύπλεξης. Δέχεται αυτοδύναμα πακέτα UDP από πολλά προγράμματα εφαρμογών και τα μεταβιβάζει στο IP για μετάδοση και ακόμη δέχεται εισερχόμενα πακέτα UDP από το IP και τα μεταβιβάζει στο κατάλληλο πρόγραμμα εφαρμογής.

Θεωρητικά όλη η διαδικασία της πολύπλεξης ή της αποπολύπλεξης ανάμεσα στο λογισμικό UDP και στα προγράμματα εφαρμογών γίνεται με τη χρήση του μηχανισμού θυρών. Ουσιαστικά κάθε πρόγραμμα εφαρμογής, πριν στείλει ένα αυτοδύναμο πακέτο, πρέπει να αποκτήσει μια θύρα πρωτοκόλλου και έναν αντίστοιχο αριθμό θύρας. Αφού γίνει αυτό, κάθε αυτοδύναμο πακέτο, που στέλνεται μέσω της θύρας από το πρόγραμμα εφαρμογής, θα έχει τον αριθμό της θύρας- που εκχωρήθηκε- στο πεδίο Θύρα προέλευσης UDP. Στη συνέχεια, όταν

επεξεργάζεται η είσοδος, το UDP δέχεται τα εισερχόμενα αυτοδύναμα πακέτα από το λογισμικό IP και τα αποπολύπλεκει με βάση τη Θύρα προορισμού UDP.

Η διαδικασία αυτή φαίνεται και στην παρακάτω εικόνα.



Ο πιο εύκολος τρόπος να αναπαραστήσουμε στο μυαλό μας μια θύρα UDP είναι με τη μορφή μιας ουράς. Όταν ένα πρόγραμμα εφαρμογής εκτελεί τις κατάλληλες ενέργειες για να χρησιμοποιήσει μία θύρα, το λειτουργικό σύστημα δημιουργεί μια εσωτερική ουρά που μπορεί να αποθηκεύσει τα εισερχόμενα μηνύματα. Συνήθως, η εφαρμογή ορίζει ή αλλάζει το μέγεθος της ουράς. Μόλις το UDP λάβει ένα μήνυμα, ελέγχει για να δει αν ο αριθμός της θύρας προορισμού ταιριάζει με κάποια από τις ενεργές θύρες. Αν αυτό δεν συμβαίνει τότε στέλνει ένα μήνυμα σφάλματος τύπου μη προσπελάσιμη θύρα και απορρίπτει το αυτοδύναμο πακέτο. Αν, όμως, βρεθεί η αντίστοιχη θύρα, τότε το UDP τοποθετεί το νέο πακέτο στη θύρα από την οποία θα μπορέσει να το προσπελάσει κάποιο πρόγραμμα εφαρμογής. Ακόμη, αν η θύρα είναι πλήρης, τότε δημιουργείται σφάλμα και το UDP απορρίπτει το εισερχόμενο πακέτο.

5.5. Υπολογισμός αθροίσματος ελέγχου UDP

Η μέθοδος που χρησιμοποιείται για τον υπολογισμό του αθροίσματος ελέγχου ορίζεται στο RFC 768 ως εξής:

Το άθροισμα ελέγχου είναι το 16μπιτο συμπλήρωμα ως προς ένα του συμπληρώματος ως προς ένα του αθροίσματος της ψευδοκεφαλίδας των

πληροφοριών από το IP header, το UDP header και τα δεδομένα συμπληρωμένα με οκτάδες μηδενικών –αν είναι απαραίτητο.

Με άλλα λόγια όλες οι 16μπιτες λέξεις αθροίζονται χρησιμοποιώντας αριθμητική συμπληρώματος ως προς ένα. Στη συνέχεια παίρνουμε το συμπλήρωμα ως προς ένα του αθροίσματος ώστε να πάρουμε την τιμή του πεδίου Άθροισμα ελέγχου UDP. Αν το αποτέλεσμα είναι μηδέν, δηλαδή και τα 16 δυαδικά ψηφία να είναι μηδέν, τότε αφού θα σταλεί σαν συμπλήρωμα ως προς ένα, θα σταλούν και τα 16 δυαδικά ψηφία με την τιμή 1.

Εδώ πρέπει να πούμε ότι η διαφορά ανάμεσα στο IPv4 και IPv6 υπάρχει στα δεδομένα που χρησιμοποιούνται για τον υπολογισμό του checksum.

5.5.1. IPv4

Όταν το UDP χρησιμοποιεί IPv4, το άθροισμα ελέγχου υπολογίζεται χρησιμοποιώντας μία ψευδοκεφαλίδα που περιέχει μερικές πληροφορίες ίδιες από την κανονική κεφαλίδα του IPv4, όμως δεν είναι ακριβώς ίδια με την κανονική κεφαλίδα του IPv4 που χρησιμοποιείται για στείλει ένα IP πακέτο.

Στο σχήμα που ακολουθεί φαίνεται η ψευδοκεφαλίδα που χρησιμοποιείται μόνο για τον υπολογισμό του αθροίσματος ελέγχου.

bits	0 – 7	8 – 15	16 – 23	24 – 31
0	Διεύθυνση προέλευσης			
32	Διεύθυνση προορισμού			
64	Μηδενικά	Πρωτόκολλο	Μήκος UDP	
96	Θύρα πηγής		Θύρα προορισμού	
128	Μήκος		Άθροισμα ελέγχου	
160+	Δεδομένα			

Οι διευθύνσεις προέλευσης και προορισμού είναι αυτές που υπάρχουν και στην κεφαλίδα του IPv4. Το πεδίο Πρωτόκολλο για UDP είναι 17 (0x11). Το πεδίο Μήκος UDP είναι το μήκος της κεφαλίδας UDP και των δεδομένων.

Για το IPv4 ο υπολογισμός του αθροίσματος ελέγχου είναι προαιρετικός. Εάν δεν χρησιμοποιείται πρέπει να τίθεται στην τιμή μηδέν.

5.5.2. IPv6

Όταν το UDP χρησιμοποιεί IPv6 ο υπολογισμός του αθροίσματος ελέγχου είναι υποχρεωτικός. Η μέθοδος που χρησιμοποιείται για τον υπολογισμό του είναι διαφορετικός και ορίζεται στο RFC 2460 ως εξής:

Οποιοδήποτε πρωτόκολλο μεταφοράς ή πρωτόκολλο ανωτέρου επιπέδου, το οποίο περιλαμβάνει τις διευθύνσεις από το IP header στον υπολογισμό του Αθροίσματος ελέγχου του πρέπει να διαφοροποιείται για να χρησιμοποιηθεί με IPv6 ώστε να συμπεριλάβει και τις 128μπιτες διευθύνσεις IPv6.

Και εδώ για τον υπολογισμό του αθροίσματος ελέγχου χρησιμοποιείται μία ψευδοκεφαλίδα παρόμοια, αλλά όχι ίδια, με την κανονική κεφαλίδα IPv6. Αυτή παρουσιάζεται στο παρακάτω σχήμα:

Η διεύθυνση προέλευσης είναι αυτή που υπάρχει και στην κεφαλίδα του IPv6. Η διεύθυνση προορισμού είναι ο τελικός προορισμός. Αν το πακέτο IPv6 δεν περιέχει κεφαλίδα routing αυτή θα είναι η διεύθυνση προορισμού στην κεφαλίδα IPv6.

Η τιμή του πεδίου Επόμενη κεφαλίδα είναι η τιμή πρωτοκόλλου για UDP, δηλαδή 17. Το πεδίο Μήκος UDP είναι το μήκος της κεφαλίδας UDP και των δεδομένων.

bits	0 – 7	8 – 15	16 – 23	24 – 31
0	Διεύθυνση προέλευσης			
32				
64				
96				
128	Διεύθυνση προορισμού			
160				
192				
224				
256	Μήκος UDP			
288	Μηδενικά		Επόμενη Κεφαλίδα	
320	Θύρα πηγής		Θύρα προορισμού	
352	Μήκος		Άθροισμα ελέγχου	
384+	Δεδομένα			

5.6. Αριθμοί θυρών UDP

Εδώ θα δούμε πως εκχωρούνται οι αριθμοί των θυρών. Θα αναφερθούμε σε αυτό το θέμα, διότι αποτελεί ένα σημαντικό πρόβλημα, αφού πριν επικοινωνήσουν δύο υπολογιστές θα πρέπει πρώτα να συμφωνήσουν σχετικά με τους αριθμούς των θυρών τους.

Υπάρχουν δύο κύριες προσεγγίσεις στο θέμα της εκχώρησης θυρών.

Η πρώτη προσέγγιση χρησιμοποιεί μια **κεντρική αρχή**, η οποία αναλαμβάνει να εκχωρεί αριθμούς θυρών ανάλογα με τις ανάγκες και να κοινοποιεί τη λίστα με όλες τις εκχωρήσεις. Στη συνέχεια, όλο το λογισμικό δημιουργείται με βάση αυτή τη λίστα. Αυτού του είδους η προσέγγιση ονομάζεται **παγκόσμια εκχώρηση** (universal assignment) και οι εκχωρήσεις θυρών που δημιουργούνται /ορίζονται από την κεντρική αρχή ονομάζονται ευρέως γνωστές εκχωρήσεις θυρών (well-known port assignments).

Η δεύτερη προσέγγιση χρησιμοποιεί **δυναμική αντιστοίχιση**. Σε αυτή την περίπτωση, οι θύρες δεν είναι παγκοσμίως γνωστές. Αντίθετα, κάθε φορά που ένα πρόγραμμα χρειάζεται μια θύρα, το λογισμικό δικτύου του εκχωρεί μία. Προκειμένου να μάθουμε τις τρέχουσες εκχωρήσεις θυρών σε κάποιον άλλον υπολογιστή πρέπει να του στείλουμε μία σχετική αίτηση. Τότε η μηχανή προορισμού απαντά δίνοντας τον σωστό αριθμό θύρας.

Οι σχεδιαστές του μοντέλου TCP/IP δεν χρησιμοποιούν ούτε την πρώτη ούτε την δεύτερη προσέγγιση αποκλειστικά. Η προσέγγιση που χρησιμοποιούν είναι μία υβριδική, η οποία εκχωρεί ορισμένους αριθμούς θυρών εκ των προτέρων, αφήνει όμως και πολλά περιθώρια για εκχώρηση αριθμών σε τοποθεσίες και προγράμματα εφαρμογών. Οι εκχωρημένοι αριθμοί θυρών ξεκινούν από χαμηλές τιμές και επεκτείνονται προς τα πάνω, αφήνοντας διαθέσιμες τις μεγαλύτερες τιμές για δυναμικές εκχωρήσεις.

Στη συνέχεια ακολουθεί ένας πίνακας όπου υπάρχουν μερικοί από τους τρέχοντες εκχωρημένους αριθμούς θυρών UDP.

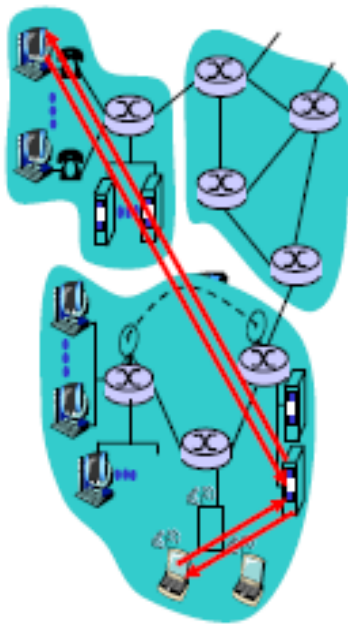
Δεκαδική τιμή	Λέξη-κλειδί	Λέξη-κλειδί UNIX	Περιγραφή
0	-	-	Δεσμευμένη
7	ECHO	echo	Αντήχηση
9	DISCARD	discard	Απόρριψη
11	USERS	systat	Ενεργοί χρήστες
13	DAYTIME	daytime	Ημέρα/Ώρα

15	-	netstat	Πρόγραμμα κατάστασης δικτύου
17	QUOTE	qotd	Απόφθεγμα της ημέρας
19	CHARGEN	chargen	Γεννήτρια χαρακτήρων
37	TIME	time	Ώρα
42	NAMESERVER	name	Διακομιστής ονομάτων υπολογιστών υπηρεσίας
43	NICNAME	who is	Ποιος είναι
53	DOMAIN	nameserver	Διακομιστής ονομάτων περιοχής
67	BOOTPS	bootps	Διακομιστής BOOTP ή DHCP
68	BOOTPC	bootpc	Πελάτης BOOTP ή DHCP
69	TFTP	tftp	Στοιχειώδες πρωτόκολλο μεταφοράς αρχείων
88	KERBEROS	Kerberos	Υπηρεσία ασφαλείας Kerberos
111	SUNRPC	sunrpc	Κλήσεις απομακρυσμένων διαδικασιών Sun
123	NTP	ntp	Πρωτόκολλο Ώρας Δικτύου
161	-	snmp	Πρωτόκολλο SNMP
162	-	snmp-trap	Παγιδεύσεις SNMP
512	-	biff	Διαταγή comsat του UNIX
513	-	who	Δαίμων rwho του UNIX
514	-	syslog	Αρχείο καταγραφής συστήματος
525	-	timed	Δαίμων Time

6. UDP server και client

6.1. Εισαγωγή στην έννοια του client-server computing

Γενικά, το client-server computing αναφέρεται σε μια βασική αλλαγή στο στυλ των υπολογιστών, την αλλαγή από τα συστήματα που βασίζονται στα μηχανήματα στα συστήματα που βασίζονται στον χρήστη.



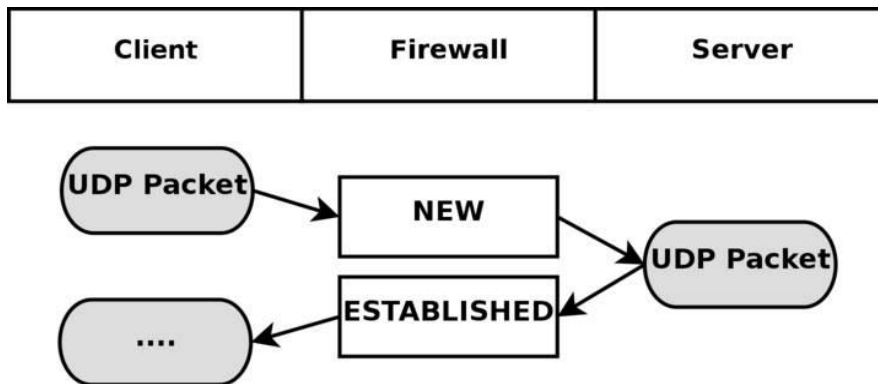
Ειδικότερα, ένα σύστημα client-server είναι ένα σύστημα στο οποίο το δίκτυο ενώνει διάφορους υπολογιστικούς πόρους, ώστε οι clients να μπορούν να ζητούν υπηρεσίες από έναν server, ο οποίος προσφέρει πληροφορίες ή επιπρόσθετη υπολογιστική ισχύ. Με άλλα λόγια, ο client θέτει μια αίτηση και ο server επιστρέφει μια ανταπόκριση ή κάνει μια σειρά από ενέργειες. Ο server μπορεί να ενεργοποιείται άμεσα για την αίτηση αυτή ή να προσθέτει την αίτηση σε μια ουρά.

Το client-server computing είναι πολύ σημαντικό, διότι επιτυγχάνει τα εξής:

- ✓ Αποτελεσματική χρήση της υπολογιστικής ισχύος.
- ✓ Μείωση του κόστους συντήρησης, δημιουργώντας συστήματα client-server που απαιτούν λιγότερη συντήρηση και κοστίζουν λιγότερο στην αναβάθμιση.
- ✓ Αύξηση της παραγωγικότητας, προσφέροντας στους χρήστες ξεκάθαρη πρόσβαση στις αναγκαίες πληροφορίες μέσω σταθερών και εύκολων στη χρήση διασυνδέσεων.
- ✓ Αύξηση της ευελιξίας και της δυνατότητας δημιουργίας συστημάτων που υποστηρίζουν πολλά περιβάλλοντα.

6.2. Βασικό client-server μοντέλο

Η πλευρά του client πρώτα στέλνει ένα μήνυμα για να καλέσει σε ετοιμότητα τον server. Από τη στιγμή που ο client και ο server έχουν επικοινωνία μεταξύ τους, ο client μπορεί να υποβάλλει την αίτησή του.



6.2.1. Client

Ο client είναι ο αιτών των υπηρεσιών. Δεν μπορεί παρά να είναι ένας υπολογιστής. Οι υπηρεσίες που ζητάει ένας client μπορεί να υπάρχουν στους ίδιους σταθμούς εργασίας ή σε απομακρυσμένους σταθμούς εργασίας που συνδέονται μεταξύ τους μέσω ενός δικτύου. Να τονίσουμε, ακόμη, ότι ο client είναι αυτός που ξεκινάει πάντα την επικοινωνία.

Μια client μηχανή πρέπει να μπορεί να κάνει τα ακόλουθα:

- ✓ Να τρέχει το λογισμικό των γραφικών διεπαφών χρηστών (GUIs).
- ✓ Να δημιουργεί τις αιτήσεις για πληροφορίες και να τις στέλνει στον server.
- ✓ Να αποθηκεύει τις επιστρεφόμενες πληροφορίες.

Αυτές οι αιτήσεις καθορίζουν πόση μνήμη χρειάζεται, ποια ταχύτητα επεξεργασίας θα μπορούσε να βελτιώσει τον χρόνο ανταπόκρισης και πόση χωρητικότητα αποθήκευσης απαιτείται.

6.2.2. Server

Ο server απαντάει στις αιτήσεις που γίνονται από τους clients. Δεν ξεκινάει τις επικοινωνίες, αλλά περιμένουν τις αιτήσεις των clients.

Τα συστατικά του server είναι πολύ απλά. Μια μηχανή server πρέπει να μπορεί να κάνει τα ακόλουθα:

- ✓ Να αποθηκεύει, να ανακτά και να προστατεύει πληροφορίες
- ✓ Να επιθεωρεί τις αιτήσεις των clients.
- ✓ Να δημιουργεί εφαρμογές διαχείρισης πληροφοριών, όπως δημιουργία αντιγράφων, ασφάλεια κ.α.
- ✓ Να διαχειρίζεται πληροφορίες.

6.2.3. Δίκτυα

Τα δίκτυα είναι τα πιο άγνωστα συστατικά στην εξίσωση των client- server. Αυτό συμβαίνει διότι τα συστήματα αυτά είναι σχεδιασμένα για να κάνουν τα δίκτυα διάφανα στον χρήστη. Επιπλέον, τα δίκτυα πρέπει να είναι αξιόπιστα. Πρέπει να μπορούν να υποστηρίξουν την επικοινωνία, να ελέγχουν σφάλματα και να ξεπερνούν αμέσως τις αποτυχίες.

Τα δίκτυα ελέγχονται από το λογισμικό λειτουργικών συστημάτων και διαχείρισης για να ελέγχουν τις υπηρεσίες επικοινωνίας του server και προστατεύουν τα προγράμματα του client και του server από το να έχουν άμεση σύνδεση μεταξύ τους. Το λογισμικό διαχείρισης εστιάζεται στην παροχή αξιόπιστων υπηρεσιών, στην ελαχιστοποίηση των προβλημάτων στο δίκτυο και στην ελαχιστοποίηση των χρόνων «πτώσης» του δικτύου.

6.3. Προγραμματισμός με UDP

Το πρωτόκολλο UDP, αν και δεν εγγυάται την μετάδοση των δεδομένων μας, έχει ορισμένα χαρακτηριστικά που διευκολύνουν τον προγραμματισμό διαδικτυακών εφαρμογών:

- Δεν υπάρχει η έννοια της σύνδεσης (connection) σε αντίθεση με το TCP. Έτσι, δεν απαιτείται η χρήση threads για εξυπηρέτηση πολλαπλών clients.
- Ένα πακέτο δεδομένων UDP αυτό-περιορίζει (self-delimits) τα δεδομένα του. Όταν το λάβουμε ξέρουμε ότι τα δεδομένα που μας έστειλαν είναι αυτά και μόνο αυτά, ούτε λιγότερα ούτε περισσότερα.

Τώρα, μιας και γνωρίζουμε την θεωρία των διαδικτυακών πρωτοκόλλων, παραθέτουμε μια σειρά εννοιών που είναι σχετικές με το UDP (να πούμε ότι είναι σχετικές και με το TCP).

- **Διευθύνσεις:**
Αποτελούνται από δύο μέρη. Το πρώτο είναι η διεύθυνση IP, πολλές φορές ονομάζεται και host ή interface αφού αντιστοιχεί μια διεύθυνση ανά δικτυακή σύνδεση του υπολογιστή. Το δεύτερο είναι ο αριθμός port. Εάν το πρόγραμμά μας παίζει τον ρόλο του server πρέπει να δεθεί σε ένα γνωστό port για να ξέρουν οι clients που θα συνδεθούν. Τυπικά μια εφαρμογή χωρίς δικαιώματα superuser μπορεί να δεθεί μόνο σε ports.
- **Μέγιστος αριθμός λαμβανόμενων δεδομένων:**

Όλες οι συναρτήσεις λήψης για το UDP δέχονται ως παράμετρο bufsize τον μέγιστο αριθμό δεδομένων που μπορούν να επιστρέψουν. Αυτό δεν σημαίνει ότι η συνάρτηση λήψης θα επιστρέψει μόνο αφού εμφανιστούν bufsize bytes δεδομένων. Μπορεί να επιστρέψει και με λιγότερα δεδομένα. Αν τα δεδομένα που έχουν φτάσει είναι περισσότερα από bufsize bytes η συνάρτηση λήψης θα επιστρέψει ακριβώς bufsize bytes. Έπειτα θα πρέπει να επαναληφθεί η κλήση της συνάρτησης και για τα υπόλοιπα.

- **Δημιουργία sockets:**
Δημιουργούμε αντικείμενα socket με την χρήση της μεθόδου socket.socket().
- **«Δέσιμο» με διεύθυνση λήψης:**
Μετά τη δημιουργία του ένα socket είναι unbound. Για να δεθεί με συγκεκριμένη διεύθυνση χρησιμοποιούμε τη μέθοδο bind().
- **Κλείσιμο socket:**
Γίνεται με την μέθοδο close().

6.4. Δείγματα απλού κώδικα

6.4.1. UDP server-client σε C

Παρακάτω θα δούμε ένα απλό παράδειγμα προγράμματος server-client σε C. Ο server τρέχει σε ένα τοπικό υπολογιστή περιμένοντας αίτηση από έναν απομακρυσμένο υπολογιστή, ο οποίος ζητάει την παρούσα ώρα του server. Ο server στην συνέχεια επιστρέφει την current ώρα του στον client, ο οποίος με την σειρά του την εμφανίζει.

Ο client στέλνει ένα datagram στον server, ο οποίος στη συνέχεια επεξεργάζεται τις πληροφορίες και επιστρέφει μια απάντηση. Ο server περιμένει το datagram από τους clients και όταν λαμβάνει ένα datagram που περιέχει το μήνυμα "GET TIME\r\n" επιστρέφει την παρούσα ώρα.

Αρχικά ανοίγουμε σύνδεση στα Windows. Στη συνέχεια ανοίγουμε ένα socket για το UDP. Μετά από αυτό φτιάχνουμε τον server.

Πρώτα καθαρίζουμε την μνήμη. Στη συνέχεια καθορίζουμε την μορφή των δεδομένων και τον αριθμό της θύρας. Ανάλογα με τις παραμέτρους που εισάγονται ο server είτε θα προσπαθήσει να πάρει την δική του IP διεύθυνση είτε καθορίζεται χειροκίνητα. Τώρα το επόμενο βήμα είναι να κάνουμε bind την διεύθυνση του server στο socket. Τώρα ο server είναι έτοιμος να «ακούσει» τα datagrams από τους clients. Μόλις ο server λάβει το datagram συγκρίνει τις πληροφορίες μέσα σε αυτό με το αλφαριθμητικό "GET TIME\r\n". Αν τα δυο

αυτά αλφαριθμητικά είναι ίδια τότε ο server επιστρέφει την ώρα. Σε κάθε άλλη περίπτωση η αίτηση απορρίπτεται σαν μη έγκυρη αίτηση.

```

/* Open windows connection */
if (WSAStartup(0x0101, &w) != 0)
{
    fprintf(stderr, "Could not open Windows connection.\n");
    exit(0);
}
/* Open a datagram socket */
sd = socket(AF_INET, SOCK_DGRAM, 0);
if (sd == INVALID_SOCKET)
{
    fprintf(stderr, "Could not create socket.\n");
    WSACleanup();
    exit(0);
}

/* Clear out server struct */
memset((void *)&server, '\0', sizeof(struct sockaddr_in));

/* Set family and port */
server.sin_family = AF_INET;
server.sin_port = htons(port_number);

/* Set address automatically if desired */
if (argc == 2)
{
    /* Get host name of this computer */
    gethostname(host_name, sizeof(host_name));
    hp = gethostbyname(host_name);

    /* Check for NULL pointer */
    if (hp == NULL)
    {
        fprintf(stderr, "Could not get host name.\n");
        closesocket(sd);
        WSACleanup();
        exit(0);
    }

    /* Assign the address */
    server.sin_addr.S_un.S_un_b.s_b1 = hp->h_addr_list[0][0];
    server.sin_addr.S_un.S_un_b.s_b2 = hp->h_addr_list[0][1];
    server.sin_addr.S_un.S_un_b.s_b3 = hp->h_addr_list[0][2];
    server.sin_addr.S_un.S_un_b.s_b4 = hp->h_addr_list[0][3];
}
/* Otherwise assign it manually */
else
{
    server.sin_addr.S_un.S_un_b.s_b1 = (unsigned char)a1;
    server.sin_addr.S_un.S_un_b.s_b2 = (unsigned char)a2;
    server.sin_addr.S_un.S_un_b.s_b3 = (unsigned char)a3;
    server.sin_addr.S_un.S_un_b.s_b4 = (unsigned char)a4;
}

/* Bind address to socket */
if (bind(sd, (struct sockaddr *)&server,

```

```

        sizeof(struct sockaddr_in)) == -1)
{
    fprintf(stderr, "Could not bind name to socket.\n");
    closesocket(sd);
    WSACleanup();
    exit(0);
}

client_length = (int)sizeof(struct sockaddr_in);

/* Receive bytes from client */
bytes_received = recvfrom(sd, buffer, BUFFER_SIZE, 0,
    (struct sockaddr *)&client, &client_length);
if (bytes_received < 0)
{
    fprintf(stderr, "Could not receive datagram.\n");
    closesocket(sd);
    WSACleanup();
    exit(0);
}

/* Check for time request */
if (strcmp(buffer, "GET TIME\r\n") == 0)
{
    /* Get current time */
    current_time = time(NULL);

    /* Send data back */
    if (sendto(sd, (char *)&current_time,
        (int)sizeof(current_time), 0,
        (struct sockaddr *)&client, client_length) !=
        (int)sizeof(current_time))
    {
        fprintf(stderr, "Error sending datagram.\n");
        closesocket(sd);
        WSACleanup();
        exit(0);
    }
}
}

```

Όσον αφορά τώρα το πρόγραμμα του client, είναι ένας απλός UDP client, ο οποίος στέλνει αίτηση στον server για να πάρει την παρούσα ώρα και μετά περιμένει να λάβει πίσω αυτή την ώρα. Η σύνδεση στα Windows αρχικά είναι ανοιχτή. Στη συνέχεια ανοίγει και το socket. Ο κώδικας είναι πολύ απλός και παρόμοιος με τον κώδικα του server.

```

/* Transmit data to get time */
server_length = sizeof(struct sockaddr_in);
if (sendto(sd, send_buffer, (int)strlen(send_buffer) + 1,
    0, (struct sockaddr *)&server, server_length) == -1)
{
    fprintf(stderr, "Error transmitting data.\n");
    closesocket(sd);
    WSACleanup();
    exit(0);
}
}

```

```

/* Receive time */
if (recvfrom(sd, (char *)&current_time,
            (int)sizeof(current_time), 0,
            (struct sockaddr *)&server,
            &server_length) < 0)
{
    fprintf(stderr, "Error receiving data.\n");
    closesocket(sd);
    WSACleanup();
    exit(0);
}

closesocket(sd);
WSACleanup();

```

6.4.2. UDP server-client σε Java

UDPServer.java

```

import java.io.*;
import java.net.*;

class UDPServer
{
    public static void main(String args[]) throws Exception
    {
        DatagramSocket serverSocket = new DatagramSocket(9876);
        byte[] receiveData = new byte[1024];
        byte[] sendData = new byte[1024];
        while(true)
        {

            DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
                serverSocket.receive(receivePacket);

            String sentence = new String( receivePacket.getData());
                System.out.println("RECEIVED: " + sentence);

            InetAddress IPAddress = receivePacket.getAddress();
                int port = receivePacket.getPort();

            String capitalizedSentence = sentence.toUpperCase();
                sendData = capitalizedSentence.getBytes();
                DatagramPacket sendPacket =
                new DatagramPacket(sendData, sendData.length,
IPAddress, port);
                serverSocket.send(sendPacket);
        }
    }
}

```

UDPClient.java

```

import java.io.*;
import java.net.*;

class UDPClient
{
    public static void main(String args[]) throws Exception
    {
        BufferedReader inFromUser =
            new BufferedReader(new InputStreamReader(System.in));
        DatagramSocket clientSocket = new DatagramSocket();

        InetAddress IPAddress = InetAddress.getByName("localhost");
        byte[] sendData = new byte[1024];
        byte[] receiveData = new byte[1024];
        String sentence = inFromUser.readLine();
        sendData = sentence.getBytes();
        DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, IPAddress, 9876);
        clientSocket.send(sendPacket);

        DatagramPacket receivePacket = new DatagramPacket(receiveData
, receiveData.length);
        clientSocket.receive(receivePacket);

        String modifiedSentence = new String(receivePacket.getData())
;
        System.out.println("FROM SERVER:" + modifiedSentence);
        clientSocket.close();
    }
}

```

7. Σύγκριση UDP με TCP

Αρχικά πρέπει να αναφέρουμε ότι και τα δύο είναι πρωτόκολλα του επιπέδου μεταφοράς της σουίτας TCP/IP και προσφέρουν διαδικτυακές υπηρεσίες για εφαρμογές και πρωτόκολλα εφαρμογών.

Η θεμελιώδης διαφορά τους είναι ότι το UDP είναι πρωτόκολλο χωρίς σύνδεση (connectionless), επειδή στέλνει τα δεδομένα χωρίς πάντα να επιτυγχάνει μια σύνδεση, ενώ το TCP είναι προσανατολισμένο στην σύνδεση, επειδή προτού καταφέρει να αρχίσει η μια διαδικασία εφαρμογής να στέλνει τα δεδομένα σε μια άλλη πρέπει και οι δύο διαδικασίες να «συμφωνήσουν» η μία με την άλλη.

Για το TCP, από τη στιγμή που εγκαθιδρυθεί μια σύνδεση με επιτυχία όλα τα δεδομένα αποστέλλονται από τον έναν υπολογιστή στον άλλον σε μορφή πακέτων χρησιμοποιώντας την σύνδεση αυτή. Τα κύρια χαρακτηριστικά του TCP είναι τα εξής:

- **Αξιοπιστία**
Το TCP χρησιμοποιεί διάφορους μηχανισμούς ούτως ώστε να διασφαλίσει ότι τα πακέτα που μεταδίδονται από τον αποστολέα θα φτάσουν σίγουρα στον παραλήπτη και στην σωστή σειρά. Οι μηχανισμοί αυτοί περιλαμβάνουν την επιβεβαίωση λήψης πακέτου από τον παραλήπτη, την επαναποστολή πακέτων που χάθηκαν και τον καθορισμό ενός timeout. Στην περίπτωση που χαθεί κάποιο πακέτο, ο αποστολέας προσπαθεί και πάλι να το στείλει. Επίσης, εάν ο παραλήπτης διαπιστώσει ότι ένα πακέτο δεν του έχει έρθει τότε θα ζητήσει από τον αποστολέα να του το στείλει ξανά.
- **Σειρά πακέτων**
Εάν δύο πακέτα αποσταλούν σε μία σύνδεση το ένα μετά το άλλο τότε το πρωτόκολλο TCP εγγυάται ότι θα φτάσουν στον παραλήπτη με την ίδια σειρά με την οποία στάλθηκαν. Στην περίπτωση που λείπει ένα πακέτο και έρθουν μελλοντικά πακέτα, τότε αυτά κατακρατούνται στον buffer μέχρις ότου φτάσει το πακέτο που λείπει. Τότε αναδιατάσσονται και εμφανίζονται με την σωστή σειρά στον παραλήπτη.
- **Βαρύτητα**
Το πρωτόκολλο TCP θεωρείται ιδιαίτερα βαρύ, δεδομένου του γεγονότος ότι χρειάζονται τουλάχιστον τρία πακέτα για την εγκαθίδρυση της σύνδεσης, πριν ακόμη μεταδοθεί οποιοδήποτε πακέτο δεδομένων. Επίσης, οι μηχανισμοί αξιοπιστίας που υλοποιεί το κάνουν ακόμη πιο βαρύ, πράγμα που έχει φυσικά σημαντικό αντίκτυπο στην ταχύτητα μετάδοσης των δεδομένων.
- **Streaming**
Τα δεδομένα σαν «stream» χωρίς τίποτα να distinguishing όπου το ένα πακέτο τελειώνει το άλλο ξεκινάει. Τα πακέτα μπορεί να είναι χωρισμένα ή ενωμένα σε μεγαλύτερα ή μικρότερα stream δεδομένων arbitrarily.

Το UDP είναι ένα πιο απλό και ελαφρύ πρωτόκολλο, στο οποίο δεν υπάρχει η έννοια της σύνδεσης. Κάθε πακέτο UDP διανύει το δίκτυο σαν μια ξεχωριστή αυτόνομη μονάδα και όχι σαν μια σειρά πακέτων σε μία σύνδεση, όπως στο TCP. Τα κύρια χαρακτηριστικά του UDP είναι τα εξής:

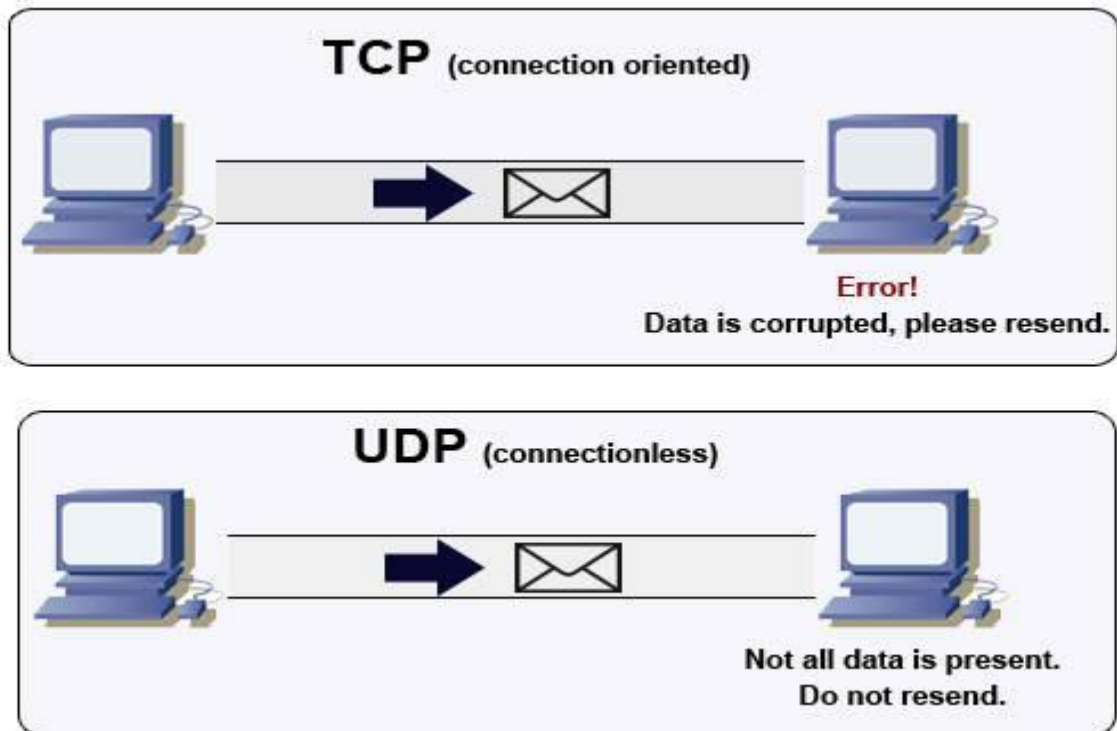
- **Αναξιόπιστο**
Κατά την αποστολή ενός πακέτου ο αποστολέας δεν είναι σε θέση να γνωρίζει εάν το πακέτο θα φτάσει σωστά στον προορισμό του ή εάν θα χαθεί μέσα στον παραλήπτη ούτε η επαναμετάδοση ενός χαμένου πακέτου.
- **Δεν υπάρχει σειρά**
Τα πακέτα UDP, σε αντίθεση με το TCP, δεν αριθμούνται και κατά συνέπεια δεν υπάρχει κάποια συγκεκριμένη σειρά με την οποία θα πρέπει να φτάσουν στον παραλήπτη.

- **Ελαφρύ**

Το πρωτόκολλο είναι αυτό καθ' αυτό ελαφρύ σε σύγκριση με το TCP διότι δεν εφαρμόζει όλους τους μηχανισμούς αξιόπιστης επικοινωνίας που υπάρχουν στο δεύτερο. Αυτό έχει ως συνέπεια να είναι αρκετά πιο γρήγορο.

- **Datagrams**

Κάθε πακέτο UDP θεωρείται ως μεμονωμένη οντότητα που θα πρέπει να μεταδοθεί ολόκληρη. Κατά συνέπεια δεν υφίσταται η έννοια της διοχέτευσης πακέτων μέσα σε ένα κανάλι/ σύνδεση.



8. Επίλογος – Περίληψη

Στο παρόν έγγραφο μιλήσαμε για τα πρωτόκολλα επικοινωνίας και αναλύσαμε δύο από αυτά, το Πρωτόκολλο Αυτοδύναμων Πακέτων Χρήστη (User Datagram Protocol - UDP) και το Πρωτόκολλο Ελέγχου Εκπομπής/ Πρωτόκολλο του Διαδικτύου (Transmission Control Protocol/ Internet Protocol, δηλαδή TCP/IP). Μιλήσαμε για τα χαρακτηριστικά τους, την δομή τους, τις ομοιότητες και τις διαφορές τους, όπως επίσης αναπτύξαμε και θέματα που έχουν να κάνουν με αυτές τις δύο έννοιες, όπως είναι το μοντέλο OSI του ISO.

Εδώ και δύο δεκαετίες περίπου έχει αναπτυχθεί μια καινούρια τεχνολογία, η οποία καθιστά δυνατή τη διασύνδεση πολλών ανόμοιων δικτύων και τη λειτουργία τους ως μια συντονισμένη μονάδα. Η τεχνολογία αυτή ονομάζεται διαδικτύωση. Είδαμε το Internet από την πλευρά του χρήστη και διαπιστώσαμε ότι αποτελείται από ένα σύνολο προγραμμάτων εφαρμογών, τα οποία προγράμματα χρησιμοποιούν τα υπάρχοντα

δίκτυα προκειμένου να επιτελέσουν χρήσιμες εργασίες επικοινωνίας. Τέτοιες εφαρμογές είναι για παράδειγμα ο Παγκόσμιος Ιστός, το Ηλεκτρονικό Ταχυδρομείο και η Μεταφορά Αρχείων. Από την πλευρά του προγραμματιστή έχουμε να πούμε ότι δημιουργεί αυτά τα προγράμματα χρησιμοποιώντας πρωτόκολλα της σουίτας TCP/IP.

Ανάλογα, λοιπόν, με την εφαρμογή που χρησιμοποιείται κάθε φορά εφαρμόζεται και διαφορετικό πρωτόκολλο (UDP ή TCP). Λόγω των χαρακτηριστικών του (αξιόπιστο, έλεγχο ροής δεδομένων, δυνατότητα επαναμετάδοσης και άφιξης στη σωστή σειρά κ.α.) το TCP φαίνεται ιδανικό για την σωστή και αξιόπιστη μεταφορά δεδομένων ανάμεσα σε δύο μηχανήματα. Το UDP, όμως, έρχεται να καλύψει την ανάγκη για μετάδοση πληροφοριών και δεδομένων γρήγορα. Υπάρχουν, δηλαδή εφαρμογές που δεν μας ενδιαφέρει τα δεδομένα να φτάσουν σωστά αλλά να φτάσουν γρήγορα. Οπότε το UDP ουσιαστικά αντικαθιστά το TCP εκεί που το TCP είναι αργό.

9. Γλωσσάρι Όρων και Συντομογραφιών

ACK

(acknowledgement)
επιβεβαίωση

DNS

(Domain Name System – Σύστημα Ονομάτων Περιοχών)

Η ηλεκτρονική κατανεμημένη βάση δεδομένων που χρησιμοποιείται για την αντιστοίχιση ονομάτων μηχανημάτων σε διευθύνσεις IP. Επιπλέον, υποστηρίζει ξεχωριστές αντιστοιχίσεις μεταξύ προορισμών αλληλογραφίας και διευθύνσεων IP.

DHCP

(Dynamic Host Configuration Protocol – Πρωτόκολλο Δυναμικής Διευθέτησης Υπολογιστών Υπηρεσίας)

Ένα πρωτόκολλο που χρησιμοποιούν οι υπολογιστές υπηρεσίας για να λάβουν όλες τις απαραίτητες πληροφορίες διευθέτησης, συμπεριλαμβανόμενης της διεύθυνσης IP.

Ethernet

Μια δημοφιλής τεχνολογία τοπικού δικτύου. Είναι ένα παθητικό ομοαξονικό καλώδιο-οι διασυνδέσεις περιέχουν όλα τα ενεργητικά στοιχεία.

HTTP

(Hypertext Transfer Protocol – Πρωτόκολλο Μεταφοράς Υπερ-κειμένου)

Το πρωτόκολλο που χρησιμοποιείται για τη μεταφορά εγγράφων Ιστού από ένα διακομιστή σε ένα φυλλομετρητή.

ICMP

(Internet Control Message Protocol – Πρωτόκολλο Μηνυμάτων Ελέγχου Internet)

Αναπόσπαστο κομμάτι του IP, το οποίο χειρίζεται τα μηνύματα σφαλμάτων και ελέγχου. Περιλαμβάνει επίσης μια αίτηση/ απάντηση αντήχησης, που χρησιμοποιείται για να ελέγξει αν ο προορισμός είναι προσπελάσιμος και αν απαντάει.

Internet

Η συλλογή δικτύων και δρομολογητών που εκτείνεται σε περισσότερες από 200 χώρες, η οποία χρησιμοποιεί πρωτόκολλα TCP/IP για να σχηματίσει ένα ενιαίο συνεργατικό εικονικό δίκτυο.

IPv4

(Internet Protocol version 4)

Η επίσημη ονομασία της τρέχουσας έκδοσης του IP

IPv6

(Internet Protocol version 6)

Η επίσημη ονομασία της τρέχουσας έκδοσης του IP

NACK

(Negative Acknowledgement- Αρνητική Επιβεβαίωση)

Αναμετάδοση πλαισίων

Το όνομα μιας συνδεσμολογίας τεχνολογίας δικτύου η οποία προσφέρεται από τις τηλεφωνικές εταιρείες.

αντιστοίχιση διεύθυνσης

Η μετάφραση μιας διεύθυνσης ανώτερου επιπέδου σε μια ισοδύναμη διεύθυνση χαμηλότερου επιπέδου

αξιόπιστη μεταφορά

Χαρακτηριστικό ενός μηχανισμού, ο οποίος εγγυάται να παραδώσει τα δεδομένα χωρίς απώλεια, χωρίς αλλοίωση, χωρίς επανάληψη και με την ίδια σειρά που αυτά στάλθηκαν ή να πληροφορήσει τον αποστολέα ότι η παράδοση δεν είναι δυνατή.

απ' άκρου εις άκρο

Χαρακτηριστικό όποιου μηχανισμού λειτουργεί μόνο στην αρχική προέλευση και τον τελικό προορισμό.

αποπολύπλεξη

Ο διαχωρισμός από μια κοινή είσοδο σε πολλές εξόδους.

ασυνδεσμική υπηρεσία

Χαρακτηριστικό οποιασδήποτε υπηρεσίας παράδοσης πακέτων που χειρίζεται κάθε πακέτο ή αυτοδύναμο πακέτο ως ξεχωριστή οντότητα και επιτρέπει στα επικοινωνούντα στοιχεία να μεταδίδουν δεδομένα πριν την αποκατάσταση επικοινωνίας.

αυτοδύναμο πακέτο IP

η βασική μονάδα πληροφοριών που μεταφέρεται σε ένα διαδίκτυο TCP/IP.

διαδίκτυο

βλ. Internet

διακομιστής

(server)

Ένα εκτελούμενο πρόγραμμα το οποίο παρέχει υπηρεσίες σε πελάτες μέσω ενός δικτύου.

διεύθυνση

Μια ακέραια τιμή που χρησιμοποιείται για τον προσδιορισμό ενός συγκεκριμένου υπολογιστή, η οποία πρέπει να εμφανίζεται σε κάθε πακέτο που αποστέλλεται στον υπολογιστή.

δρομολογητής

(router)

Ένας αποκλειστικός υπολογιστής ειδικού σκοπού που συνδέεται σε δύο ή περισσότερα δίκτυα και προωθεί πακέτα από το ένα στο άλλο.

έλεγχος ροής

Έλεγχος της ταχύτητας με την οποία οι υπολογιστές υπηρεσίας ή οι δρομολογητές εισάγουν πακέτα σε ένα δίκτυο ή διαδίκτυο, συνήθως για να αποφύγουν τη συμφόρηση.

ενθυλάκωση

Η τεχνική που χρησιμοποιείται από διαστρωματωμένα πρωτόκολλα, στην οποία ένα πρωτόκολλο χαμηλότερου επιπέδου δέχεται ένα μήνυμα από ένα πρωτόκολλο υψηλότερου επιπέδου και το τοποθετεί στο τμήμα δεδομένων του πλαισίου χαμηλού επιπέδου. Η ενθυλάκωση σημαίνει ότι τα αυτοδύναμα πακέτα που ταξιδεύουν σε ένα φυσικό δίκτυο έχουν μια ακολουθία κεφαλίδων, όπου η πρώτη κεφαλίδα προέρχεται από το πλαίσιο φυσικού δικτύου, η δεύτερη από το IP κ.ο.κ.

θύρα πρωτοκόλλου

Η αφαιρετική έννοια που χρησιμοποιούν τα πρωτόκολλα μεταφοράς TCP/IP για να κάνουν διάκριση μεταξύ πολλών προορισμών σε ένα συγκεκριμένο υπολογιστή υπηρεσίας.

κεφαλίδα

Πληροφορίες στην αρχή ενός πακέτου ή μηνύματος, οι οποίες περιγράφουν τα περιεχόμενα και καθορίζουν ένα προορισμό.

πακέτο

Χρησιμοποιείται άτυπα για αναφορά σε οποιοδήποτε μικρό μπλοκ δεδομένων που αποστέλλεται σε ένα δίκτυο μεταγωγής πακέτων.

παράδοση βέλτιστης προσπάθειας

(best-effort delivery)

Χαρακτηριστικό των τεχνολογιών δικτύου που δεν παρέχουν αξιοπιστία στα επίπεδα σύνδεσης. Το πρωτόκολλο IP λειτουργεί σωστά σε υλικό παράδοσης βέλτιστης προσπάθειας, επειδή το IP δεν θεωρεί ότι το υποκείμενο δίκτυο παρέχει αξιοπιστία. Το πρωτόκολλο UDP παρέχει υπηρεσίες βέλτιστης προσπάθειας για τα προγράμματα εφαρμογών.

ψευδοκεφαλίδα

Πληροφορίες διευθύνσεων IP προέλευσης και προορισμού, οι οποίες αποστέλλονται στην κεφαλίδα IP, αλλά συμπεριλαμβάνονται στο άθροισμα ελέγχου TCP ή UDP.

10. Βιβλιογραφία

Βιβλία

DOUGLAS E. COMER, Διαδίκτυα με TCP/IP ΑΡΧΕΣ, ΠΡΩΤΟΚΟΛΛΑ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ, ΤΕΤΑΡΤΗ ΑΜΕΡΙΚΑΝΙΚΗ ΕΚΔΟΣΗ, Εκδόσεις Κλειδάριθμος, 2001

PATRIC CECCARELLI- CHRISTINA FAULKNER, Δίκτυα Υπολογιστών-Εισαγωγή στη Σύγχρονη Τεχνολογία, Εκδόσεις Μ.Γκιούρδας, 2004

ANDREW S. TANENBAUM, Δίκτυα Υπολογιστών, Τέταρτη αμερικάνικη έκδοση, Εκδόσεις Κλειδάριθμος, 2003

Πανεπιστημιακές Σημειώσεις, Δίκτυα Δημόσιας Χρήσης και Διασύνδεση Δικτύων, Χρήστος Ι. Μπούρας, 2008

Ιστοσελίδες

<http://www.filaderlis.com/ebooks/ComputerNetworks-gr.pdf>

http://web.teipir.gr/new/ecs/pelab_1/tcp/inter2.htm

<http://www.digitalnews.gr/8097/%CF%84%CE%BF-%CE%BC%CE%BF%CE%BD%CF%84%CE%AD%CE%BB%CE%BF-%CE%B1%CE%BD%CE%B1%CF%86%CE%BF%CF%81%CE%AC%CF%82-osi>

http://www.diffen.com/difference/TCP_vs_UDP

RFCs

RFC 1180, TCP/IP Tutorial

RFC 791, Internet Protocol

RFC 793, Transmission Control Protocol

RFC 768, User datagram Protocol

RFC 768, UDP checksum